MINISTRY OF EDUCATION AND TRAINING

Capstone Project Document

System for Testing Hardware Device

	Nguyễn Viết Cường	SE02203
	Trần Quang Minh	SE02142
Group Members	Trần Nam Lâm	SE02048
	Nguyễn Khắc Hoàng	SE02412
	Nguyễn Mạnh Hùng	SE02087

Supervisor Mr. Phan Duy Hùng

Capstone Project Code STHD

Hanoi, December 19, 2014

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the competitions of our project work. Without them, this project would have been done with much more obstacles than it was.

Without great support of Prof. Phan Duy Hung it would be not possible to complete this project. Prof. Phan Duy Hung has always been very responsive in providing necessary information, and without their generous support project would lack in accurate information on current developments. Thanks to all of his valuable advices and guidance's, we can finish our project on time and meets all requirements. His enthusiasm is also the biggest motivations for us to continue finishing project at difficult time.

And last but not least, we would like to thank to our family and friends for their constant encouragement and support throughout this project.

Table of Contents

Table of Contents 2

ACRONYMS & DEFINITIONS 7

CHAPTER 1: INTRODUCTION 9

1.1 Idea of Capstone Project 9

1.2 The People 9

1.3 Project Information 10

1.4 Scope 10

1.5 Equipment to Be Used 11

1.6 Literature Review 18

1.6.1 Existing system 18

1.6.2 Limitations of the existing system 20

1.7 Reference 21

CHAPTER 2: PROJECT MANAGEMENT PLAN 22

2.1 Problem Definition 22

2.1.1 Project Overview 22

2.1.1.1 The Proposals and Boundaries of System 22

2.1.1.2 Development Environment 22

2.2 Project Organization 22

2.2.1 Software Process Model 22

2.2.2 Roles and Responsibilities 24

2.2.3 Tools and Techniques 24

*2.2.*3.1 Software and Tools 24

*2.2.*3.2 Hardware 25

2.3 Project Management Plan 26

2.3.1 Tasks 26
2.3.2 All Meeting Minutes 27
2.4 Risk Management Plan 28
2.4.1 Risk Identification 28
2.4.2 Risk control 30
2.5 Configuration Management 30
2.5.1 Introduction 30
2.5.2 Roles and Responsibilities 30
2.5.3 Configuration management process 31
2.5.3.1 Project Infrastructure 31
2.5.3.2 Directory structure 31
2.5.3.3 Baseline Procedure 33
2.5.3.3.1 For Document 33
2.5.3.3.2 For Source Code 34
2.6 Coding Convention 34
2.7 Communication Plan 35
CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATIONS (SRS) 36
3.1 User Requirement Specification 36
3.2 System Requirement Specification (Specific Requirements) 36
3.2.1 External Interface Requirements 36
3.2.1.1 User Interfaces 36
3.2.1.2 Hardware Interfaces 37
3.2.1.3 Software Interfaces 38
3.2.1.4 Communications Protocol 38
3.2.2 Functional Requirements 39

3.2.2.1 Use Case Diagram 39
3.2.2.2 Use Case Specification 40
3.2.3 Non-Functional Requirements 46
3.2.3.1 Usability 46
3.2.3.2 Reliability 46
3.2.3.3 Security 46
3.2.3.4 Maintainability 46
3.2.3.5 Delivery 46
CHAPTER 4: IMPLEMENTATION 47
4.1 Proposed System Architecture 47
4.1.1 Flow Chart 47
4.2 Analysis and Selection of Tool, Devices 51
4.2.1 Software 51
4.2.1.1 Arduino Development 51
4.2.1.2 Design and coding tool 51
4.2.2 Hardware 53
4.2.2.1 Arduino Uno 53
4.2.2.2 HC-05 Serial Port Bluetooth Module 58
4.2.2.3 DHT22 temperature-humidity sensor 60
4.2.2.4 4-Channel 5V Relay Module 61
4.3 Circuit Design 63
4.3.1 Circuit for Fox-301A System 63
4.3.2 Circuit for PID System 65
4.4 Algorithm and Solution Analysis 66
4.4.1 Introduction PID Controller 66

4.4.2 Using Fox-301A on System....68
4.4.3 Using PID on System....68
4.5 Interface Design....69
4.5.1 Interface for Box....69
4.5.2 Application Interface....70
CHAPTER 5: SYSTEM IMPLEMENTATION & TEST (SIT)....72
5.1 Introduction....72
5.1.1 Test Approach....73
5.1.1.1 Test type....73
5.1.1.2 Test levels....73
5.2 Test plan....73
5.2.1 Features to be tested....73
5.2.2 Test Schedule....74
5.2.3 Test Strategy....75
5.2.4 Testing Tools and Environment....76
5.3 Test Cases....76
5.3.1 Android Application....76
5.3.2 Embedded System....80
5.4 Test Results....81
CHAPTER 6: SYSTEM USER'S MANUAL....82
6.1. Installation Guide....82
6.1.1 Environment....82
6.1.2 Installation....82
6.1.2.1 Install Eclipse....82
6.1.2.1 Install JDK 8....82

6.2 User’s Guide 83
6.2.1 Paired Screen 83
6.2.1 Main Screen 84

ACRONYMS & DEFINITIONS

No.	Acronym	Definition
1	AOM	Application On Mobile
2	COM	Communication Port
3	CPU	Central Processing Unit
4	EEPROM	Electrically Erasable Programmable Read-Only Memory
5	GPS	Global Positioning System
6	GPU	Graphics Processing Unit
7	HDD	Hard Disk Drive
8	Hum	Humidity
9	IDE	Integrated Development Environment
10	LED	Light-Emitting Diode
11	MCU	Multipoint Control Unit
12	OS	Operating System
13	OS	Operating System
14	PC	Personal Computer
15	PID	Proportional Integral Derivative
16	PM	Project Manager
17	RAM	Random Access Memory
18	ROM	Read Only Memory
19	SRAM	Static Random Access Memory
20	SRS	Software Requirement Specification
21	STHD	System for testing Hardware Device
22	Temp	Temperature

23	UART	Universal Asynchronous Receiver/Transmitter
24	UI	User Interface
25	USB	Universal Serial Bus

CHAPTER 1: INTRODUCTION

1.1 Idea of Capstone Project

Two environmental factors effects to product quality are humidity and temperature. Humidity and temperature factor affect the life of the equipment from machines to the high-tech products such as computer, smartphone. Humidity is a constant threat to production efficiency and product quality and temperature is a constant threat to performance and product life. Product must be tested in the different humidity and temperature environment before using is important. To avoid such damages, product quality must be confirmed before a product is put on the market. Environmental testing not only confirms quality through such tests as simulation testing and product life testing, it also can truly be called the indispensable prerequisite to quality assurance.

Thus, our idea is to create the chamber to test hardware devices. The system will use sensor to measure the temperature and humidity continuously in the chamber. Then information that is collected by sensor is sent to the Arduino. The Arduino will send information to Android device and it will be displayed on mobile's screen. User can use Android device to change the temperature and humidity in chamber.

1.2 The People

Supervisor:

Full name	Phone	Email	Title
Phan Duy Hùng	0975597339	Hungpd@fpt.edu.vn	Supervisor

Table 1.1: Supervisor information

Team members:

No.	Full name	Student code	Email	Role in Group
1	Nguyễn Viết Cường	SE02203	*Cuongnvse02043@fpt.edu.vn*	Leader
2	Nguyễn Mạnh Hùng	SE02087	*Hungnmse02087@fpt.edu.vn*	Developer
3	Trần Nam Lâm	SE02048	*Lamtnse02048@fpt.edu.vn*	Developer
4	Trần Quang Minh	SE02142	*Minhtqse02142@fpt.edu.vn*	Reporter
5	Nguyễn Khắc Hoàng	SE02412	*Hoangnkse02412@fpt.edu.vn*	Tester

Table 1.2: Team members' information

1.3 Project Information

- Project name: *System for testing Hardware Device*
- Project code: *STHD*
- Product type: *Embedded System*
- Timeline: *12/09/2014 to 19/12/2014*

1.4 Scope

My team is trying to create a test environment that these test chambers provide an economical and space-saving solution for cramped labs needing to do a variety of temperature and humidity testing. The final product must be satisfied the following specifications:

❖ Hardware:

- Temperature/humidity sensor: It is used to measure temperature and humidity in chamber.
- The Arduino Uno: it can be used to develop interactive objects, taking inputs from a variety of sensors, and controlling a variety of relays and Bluetooth.
- Bluetooth module: it is used to communicate the Arduino to Android smartphone.

- Android Smartphone: display temperature/ humidity and control the system via Bluetooth module.
- Fox 301A: display temperature/ humidity and control the system.

❖ Chamber: Put devices in case.

❖ Main functions: Create a test environment for hardware devices.

❖ Communication between devices :

➢ Connect the Arduino to relay module, sensor and Bluetooth module.

➢ Connect the Arduino to Android Smartphone via Bluetooth.

1.5 Equipment to Be Used

➢ ***FOX-301A Programmable Humidity and Temperature Controller:***

Highlights & details:

- Humidity/temperature sensor with 3m cable
- Temperature range -55°C to 99°C
- Humidity range 0% to 100%
- 4 relay outputs (2 humidity, 2 temperature)
- Clear digital display

Description:

The FOX 301 humidity and temperature controller allows climate control of a living space, food storage space, greenhouse or terrarium in the simplest possible way, using just one device. Two temperature relays and two humidity relays are available. Typical application in the terrarium: Temperature relay one controls a heating filament, temperature relay 2 controls a fan and a shade. Humidity relay 1 controls an air humidifier or a solenoid valve, humidity relay 2 also controls the fan. The device is set up for using the DS4000 and NTC sensors.

Figure 1.1: FOX-301A Programmable Humidity and Temperature Controller

Specifications:

DESCRIPTION	FOX-301A
External Size(mm)	72(W) X 72(H) X 110(D)
Operating Range	-55.0~99.9 °C / 0~100% RH
Input Sensor	NTC / DS4000 (wire length : 3M)
Display Accuracy	Temp: ±1% rdg ± 1 digit Humidity : ±1% rdg ± 1 digit
Output	4 relay (4a) 250 VAC 2A 1R:Temp, 2R: Temp 3R:Humi, 4R: Humi
Control Method	ON / OFF
Cutting Size(mm)	68(W) X 68(H)
Power Supply	100~240VAC 50/60Hz
Programming	℃ Set / % Set / Up /Down Key
Ambient Temp/Humi	0~55 °C (32~131 °F), 35~80% RH
Weight(Include Box)	480g
Functions	control concurrently of temperature & humidity correction of the present temperature & humidity

- ***Arduino:***

Description:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter

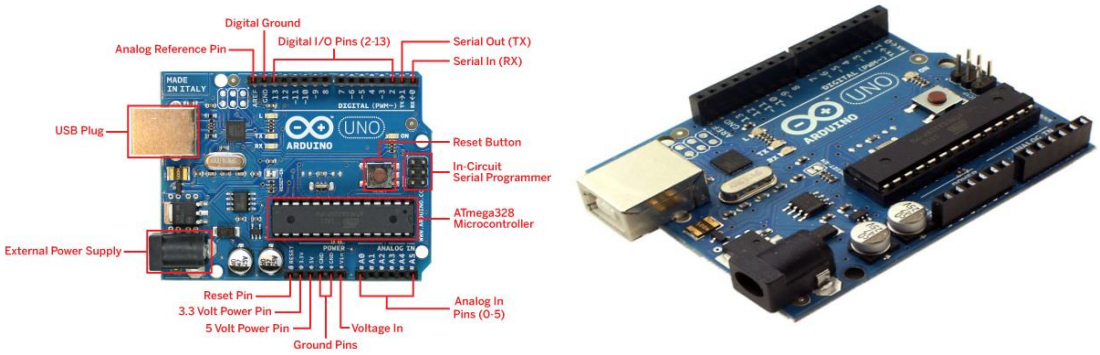


Figure 1.2: Arduino Uno

Specifications:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

- ***HC-05 Serial Port Bluetooth Module:***

Description:

Serial port Bluetooth, Drop-in replacement for wired serial connections, transparent usage. You can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project and etc.

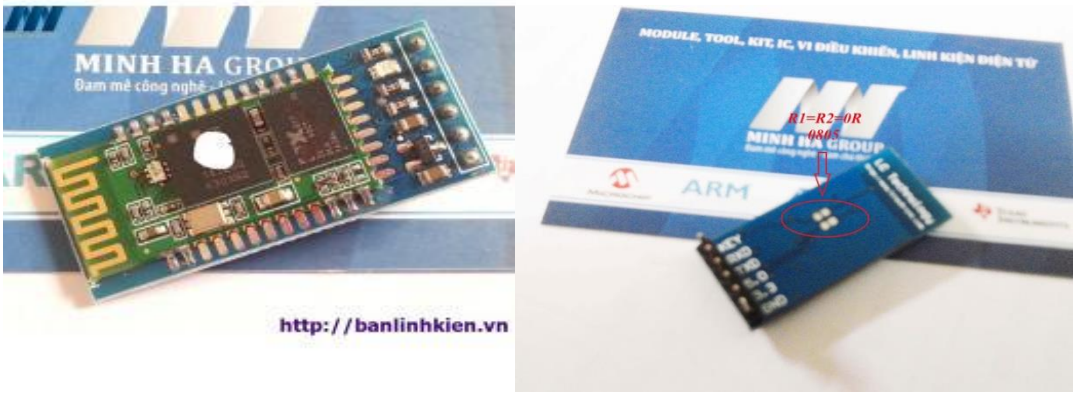


Figure 1.9: HC-05 Serial Port Bluetooth Module

- ***Case:***

  The case may be made of durable materials such as wood and metal. Size's case must be suitable enough to put all devices in case.

Figure 1.4: The case

- ***Humidifier Machine:***

  This machine uses water in humidifier chamber to produce humidity.

Figure 1.5: Humidifier machine

Specifications:

Rating Voltage	220V
Frequency	60Hz
Power	30W
Water Tank Capacity	Approz.1.0 Gallon / 3.78 Liter
Output Per 24 Hours	Approz.2.1 Gallon / 8 Liter
Run Time Per Tank	Min. 11 Hours
Humidifying Capacity	Up to 500sq.ft. / 46 m^2

Noise	< 36.5dB(A)
Maximum Evaporating Rate	280ml/hr
Maximum room size for	100 m^2
Packing Size	250x250x265mm

- ***The Light:***

This light will provide a necessary heat for the test space. With capacity is 100W, the supply of heat will be done quickly.

Figure 1.6: The light

Specifications:

Rating Voltage	220V
Power	100W

- ***Mini Fan:***

When we want to change the humidity, temperature in the test space, we will use fans is placed at the input to sucking inside moisture. Another fans always run, it is placed in the center of the barrel to mix the air inside the barrel helps to temperature and humidity at different locations in chamber is the same.

Figure 1.7: Mini Fan

Specifications:

Name	SUNON
Model	KD1206PTS1
Power	DC 12V – 2.16W
Size	50x50mm

- ***Adapter:***

It is used to convert the power supply from 220V AC to 12V DC.

Figure 1.8: Adapter

Specifications:

Name	BJF AC/DC ADAPTER
Model	BJF-DY6858
Input	AC100-240V~ 50/60Hz 0.5Amax
Output	DC12V-2A 24W Max

- ***Other equipment:***

  Wirings, casings for wirings, switches, connectors, air duct…..

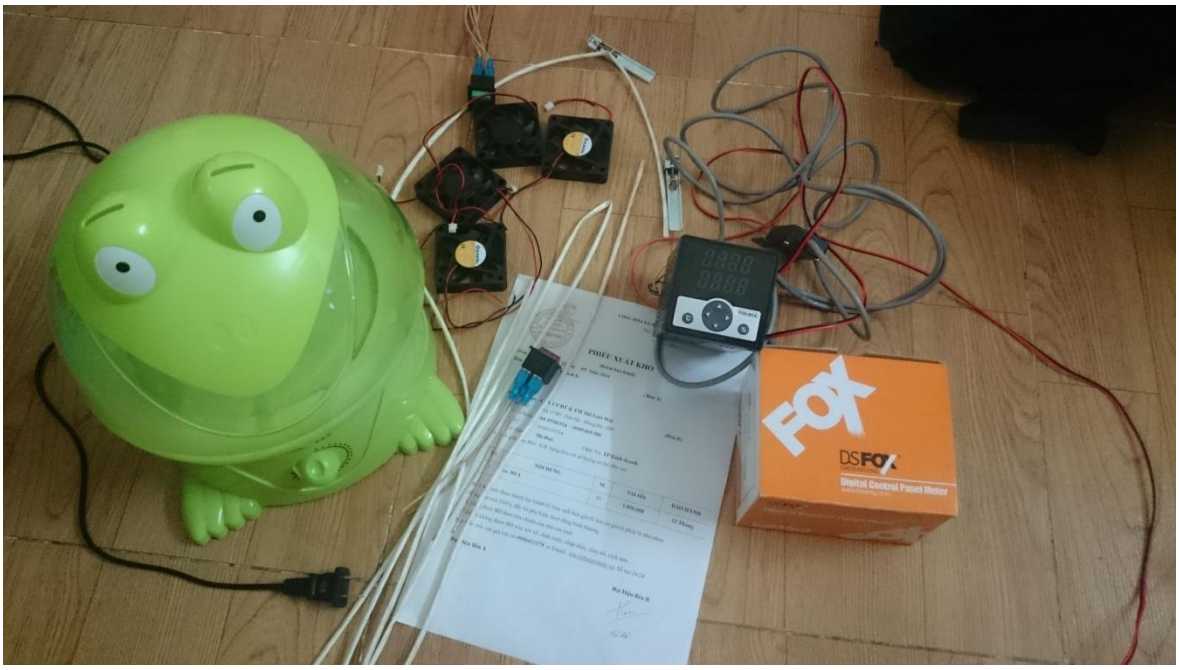


Figure 1.9: Other components

1.6 Literature Review

1.6.1 Existing system

- ***Bench top BTL-433***

ESPEC has the equipment and expertise to meet and exceed testing requirements. Criterion Temperature & Humidity Chambers provides an economical and space-saving solution for variety of testing requirements.

Features:

- Stainless steel exterior and durable thermoformed plastic door
- Full thermal break around doorframe
- One 50mm (2") diameter cable port on left, with flexible silicone plug

- Three levels of overheat protection
- 115V main power for BTL-433 model
- High performance and quiet operation (65 dBa)
- Easy lift-off service panels

Watlow F4 Programmer/Controller:

- Popular PID controller for test chambers
- Stores 40 different profiles (total 256 steps)
- Step types include: ramp, soak, jump/repeat, auto-start, and end
- RS-232C Modbus interface (free standard software)

Figure 1.10: Benchtop BTL-433

- ***CP COM Egg Incubator:***

  Egg Incubator is similar to our product. It can be created environment to use to

This system can changed temperature and humidity. With fuzzy intelligent controlling system, it is easy to operate and sterilize because trolley is movable.

Features:

- Digital temperature humidity display. Auto-temperature, auto-humidity controlling system.
- Temperature controlling precision: 0.1
- Alarm over and under heating lack.
- Ventilation by vertical air moving fan controlled by an electrical motor.
- Heating with heater controlled by electronic system with probe and display.

Figure 1.11: CP COM Egg Incubator

1.6.2 Limitations of the existing system

According to the research above, we found out that the existing systems has some inconvenient such as:

- Expensive for an exporting device.
- Shortages the system create temperature and humidity environment to test device in Viet Nam.

1.7 Reference

- http://shoptretho.com.vn/may-tao-hoi-am-magic-one-mg002-hinh-chu-ech
- http://tudonghoa24.com/--t17283.html
- http://enbac.com/Ha-Noi/Ky-thuat-so/p164209/Bo-do-va-dieu-khien-nhiet-am-Analog-Fox-300A.html
- http://banlinhkien.vn/goods-3169-cam-bien-nhiet-do-va-do-am-dht21.html#.VBzptPl_t2k
- http://banlinhkien.vn/goods-848-module-relay-2-kenh-5v-220v-10a.html#.VBznOvl_t2k

CHAPTER 2: PROJECT MANAGEMENT PLAN

2.1 Problem Definition

2.1.1 Project Overview

2.1.1.1 The Proposals and Boundaries of System

The system provides the main functions:

- Auto-Temperature/Humidity
- Display information value of temperature/humidity.

2.1.1.2 Development Environment

Below are some requirement about software and hardware:

Hardware requirement:

- Personal computer for developing with minimum configuration: CPU: Intel core i3, RAM: 4GB, HDD: 80GB
- Smart Phone Android CPU & GPU: Qualcomm APQ8064 Snapdragon Quad-core 1.7 GHz Krait, Adreno 320. Memory: RAM 2GB , ROM 32GB
- Arduino Uno: Microcontroller: ATmega328, Operating Voltage: 5V, Flash Memory 32 KB, SRAM: 2KB, EEPROM: 1KB, Clock Speed: 16MHz.

Software requirement:

- OS: Windows
- OS: Android 4.4.2
- VMware Workstation 8.0.0
- Windows 7/8

2.2 Project Organization

2.2.1 Software Process Model

The process model used for this developing this project is the Iterative and incremental software development process model. The software will be developed and updated continuously.

Iterative and incremental development is a process that grows a system feature by feature during self-contained cycles of analysis, design, and development, testing that end in the production of a stable, fully integrated and tested, partially complete system that incorporates all of the features of all previous iterations.

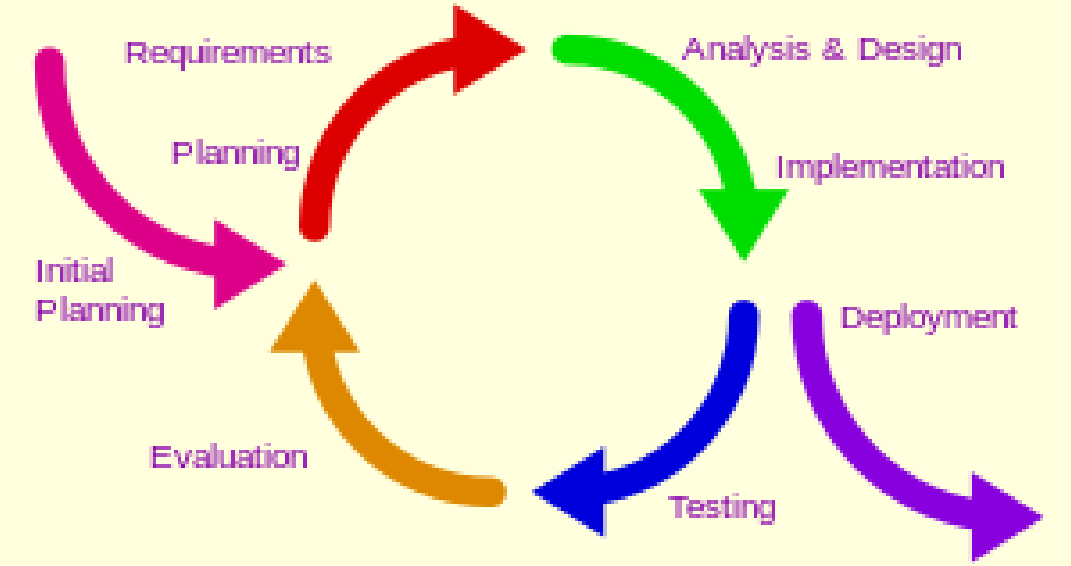


Figure 2.1: Iterative and incremental model.

We select this model because of the following advantages:

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

2.2.2 Roles and Responsibilities

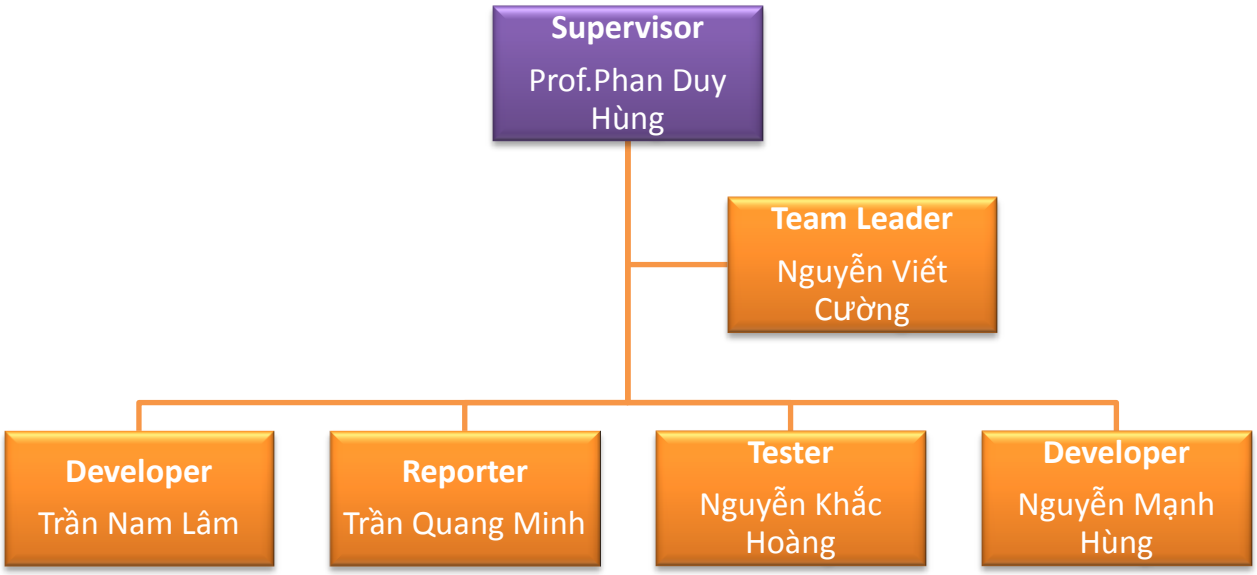


Figure 2.2: Group's roles and responsibilities

2.2.3 Tools and Techniques

*2.2.*3.1 Software and Tools

No	Software and Tools	Description
1	Google Sketch Up Make 2014 Version 14.0.4900	3D modeling computer program for applications such as interior design.
2	Altium Designer Version 9.3.1.19182	Schematic and PCB printed board design tool
3	Proteus Designer Version 12.0.0.58849	It is a Virtual System Modeling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs
4	Eclipse Juno	Eclipse can be used to develop

	Version 23.0.2.1259578	applications.
5	Microsoft Office 2010 Plus	Software is used to write report.
6	Microsoft Office Project 2007	Software is used to create plan.
7	TortoiseSVN Version 1.8.8.25755	It helps programmers manage different versions of the source code for their programs.

2.2.3.2 Hardware

No	Hardware	Description
1	Arduino	Microcontroller
2	DHT22 temperature and humidity sensor	It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin.
3	4-Channel 5V Relay Module	It is able to control various appliances, and other equipment with large current.
4	Bluetooth Module	It is used to connect Arduino with Android devices.
5	Android device	It is used to control the system.

2.3 Project Management Plan

2.3.1 Tasks

A Capstone Project	**90 days?**	**Mon 8/25/14**	**Fri 12/26/14**		
1 Project Initialization	**21 days**	**Mon 8/25/14**	**Mon 9/22/14**		
1.1 Discussing & choosing Project's ideal	6 days	Mon 8/25/14	Mon 9/1/14		Team
1.2 Lituratural Reviews	**5 days**	**Wed 9/3/14**	**Tue 9/9/14**		
1.2.1 Existing Product	2 days	Wed 9/3/14	Thu 9/4/14	3	MinhTQ
1.2.2 Existing solution and method	2 days	Wed 9/3/14	Thu 9/4/14		MinhTQ
1.2.3 Improment Object	2 days	Wed 9/3/14	Thu 9/4/14		LamTN
1.2.4 Report No.1 Composing	2 days	Mon 9/8/14	Tue 9/9/14		Team
1.3 Requirement Specification	4 days	Wed 9/10/14	Mon 9/15/14	4	Team
1.4 Bying devices	5 days	Mon 9/8/14	Fri 9/12/14		CuongNV,LamTN,I
1.5 Set the system up	1 day	Tue 9/16/14	Tue 9/16/14	9	Team
1.4 Studying and Training	7 days	Fri 9/12/14	Mon 9/22/14		Team
1.5 Deliverable: Report No1	1 day	Mon 9/22/14	Mon 9/22/14	4,9	CuongNV
2 Project Planing	**17 days**	**Tue 9/23/14**	**Wed 10/15/14**		
2.1 Project Planing making	**3 days**	**Tue 9/23/14**	**Thu 9/25/14**		
2.1.1 Project Abstract and OverView	**3 days**	**Tue 9/23/14**	**Thu 9/25/14**	**13**	
2.1.2 Tasks Speccifying	1 day	Tue 9/23/14	Tue 9/23/14		MinhTQ
2.1.3 Assignments and Timetable	2 days	Tue 9/23/14	Wed 9/24/14		CuongNV,HoangN
2.14 Risk Identification	1 day	Tue 9/23/14	Tue 9/23/14		HungNM
2.1.5 Report No.2 Composing	2 days	Wed 9/24/14	Thu 9/25/14	19	Team
2.2 Requirement Analysis	5 days	Thu 9/25/14	Wed 10/1/14		Team
2.3 Tools and Devices Analyzing and choosing	4 days	Thu 10/2/14	Tue 10/7/14		LamTN,CuongNV
2.4 Study and Training	10 days	Thu 10/2/14	Wed 10/15/14		Team
2.5 Deliverable: Report No.2 & No.3	1 day	Wed 10/15/14	Wed 10/15/14	16,21	CuongNV
3. Project Implementation	**58 days?**	**Tue 9/23/14**	**Thu 12/11/14**		
3.1 Tools and devices Analysing	1 day?	Thu 10/16/14	Thu 10/16/14	24	Team
3.2 Designning	**27 days?**	**Thu 10/2/14**	**Fri 11/7/14**		
3.2.1 Project Architechtual Designing	2 days	Fri 10/17/14	Mon 10/20/14		MinhTQ
3.2.2 Solution and Algorithms	15 days	Mon 10/20/14	Fri 11/7/14		Team
3.2.3 Circuit Designing	2 days	Fri 10/17/14	Mon 10/20/14		HungNM
3.2.4 Interface Designing	5 days	Fri 10/17/14	Thu 10/23/14		HoangNK
3.2.5 Case Designning	1 day?	Thu 10/2/14	Thu 10/2/14		Team

33	**3.3 Coding**	**57 days**	**Tue 9/23/14**	**Wed 12/10/14**	**12**	
34	**3.3.1 Application on Android smartph**	**56 days**	**Tue 9/23/14**	**Tue 12/9/14**		**LamTN,HoangNK**
35	3.3.1.1 Develop UI	10 days	Tue 9/23/14	Mon 10/6/14		HoangNK
36	3.3.1.2 Develop Functions	20 days	Tue 10/7/14	Mon 11/3/14	35	LamTN
37	3.3.1.2 Update UI & Functions	15 days	Wed 11/19/14	Tue 12/9/14	35,52	LamTN
38	3.3.1.4 Fix code	6 days	Tue 11/11/14	Tue 11/18/14	48	LamTN
39	**3.3.2 Arduino on system**	**57 days**	**Tue 9/23/14**	**Wed 12/10/14**		**Team**
40	3.3.2.1 Develop Functions	15 days	Tue 9/23/14	Mon 10/13/14		LamTN,HungNM
41	3.3.2.2 Fix code	7 days	Tue 11/11/14	Wed 11/19/14	49	LamTN
42	3.3.2.3 Update Code Functions	15 days	Thu 11/20/14	Wed 12/10/14	41	LamTN,HungNM
43	**3.4 Testing & Updating**	**46 days?**	**Tue 10/7/14**	**Tue 12/9/14**		
44	3.4.1 Create Test Cases	2 days	Mon 11/24/14	Tue 11/25/14		MinhTQ
45	**3.4.2 Unit Testing**	**25 days?**	**Tue 10/7/14**	**Mon 11/10/14**	**12**	**LamTN,HungNM**
46	3.4.2 Verify UI	5 days	Tue 10/7/14	Mon 10/13/14	35	Team
47	**3.4.3 Verify Code Functions**	**5 days**	**Tue 11/4/14**	**Mon 11/10/14**	**36**	
48	3.4.3.1 Verify App Code	5 days	Tue 11/4/14	Mon 11/10/14	36	LamTN,HungNM,H
49	3.4.3.1 Verify Arduino Code	5 days	Tue 11/4/14	Mon 11/10/14	40	LamTN,HungNM,H
50	**3.4.3 Updating**	**4 days?**	**Tue 10/7/14**	**Fri 10/10/14**		
51	3.4.3.1 Review UI	3 days	Tue 10/7/14	Thu 10/9/14	35	Team
52	3.4.3.1 Update UI	1 day?	Fri 10/10/14	Fri 10/10/14	51	Team
53	3.4.4 Modulers Testing	10 days	Wed 11/26/14	Tue 12/9/14	44	Team
54	3.4.5 Intergration Testing	10 days	Wed 11/26/14	Tue 12/9/14	44	Team
55	3.5 Deliverable: Report No.4 & Report No.5	1 day	Wed 12/10/14	Wed 12/10/14	43	CuongNV
56	3.6 Deliverable: Report No. 6	1 day	Thu 12/11/14	Thu 12/11/14	55	CuongNV
57	**4 Project Releases and Closing**	**14 days**	**Tue 12/9/14**	**Fri 12/26/14**		
58	4.1 Packaging	2 days	Tue 12/9/14	Wed 12/10/14		MinhTQ,LamTN
59	4.2 Final Report review	3 days	Fri 12/12/14	Tue 12/16/14	56	MinhTQ,LamTN,Cu
60	4.4 Presentation	1 day	Fri 12/26/14	Fri 12/26/14		Team

Figure 2.3: Task Plan

2.3.2 All Meeting Minutes

Subject	Capstone Project Daily Meeting	**Date**	09/03/2014
Facilitator		**Time**	$20^h – 22^h$
Location	Project meeting room	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		
Content	1. Searching hardware and device. 2. Planning group job and individual job.		
Subject	Capstone Project Daily Meeting	**Date**	09/11/2014
Facilitator		**Time**	$8^h – 17^h$
Location	Project meeting room	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		
Content	1. Develop a prototype.		
Subject	Capstone Project Daily Meeting	**Date**	09/29/2014
Facilitator		**Time**	$8^h – 17^h$
Location	Project meeting room	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		
Content	1. Implementing report 1 + report 2.		
Subject	Capstone Project Daily Meeting	**Date**	10/07/2014
Facilitator		**Time**	$8^h – 17^h$
Location	Project meeting room	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		
Content	1. Implementing report 3.		
Subject	Capstone Project Daily Meeting	**Date**	10/24/2014
Facilitator		**Time**	$8^h – 17^h$
Location	Project meeting room	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		

Content	1. Buying device and ordering box system. 2. Implementing report 4. 3. Design UI on mobile.		
Subject	Capstone Project Daily Meeting	**Date**	11/15/2014
Facilitator		**Time**	8^h – 17^h
Location	FPT library	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		
Content	1. Implementing report 5 + 6 2. Discussion PID algorithm and solution for system		
Subject	Capstone Project Daily Meeting	**Date**	12/05/2014
Facilitator		**Time**	8^h – 17^h
Location	Project meeting room	**Scribe**	CuongNV
Attendees	CuongNV(Team Leader), LamTN, MinhTQ, HoangNK, HungNM		
Content	1. Testing system and fixing bugs. 2. Discussion optimal solution for system.		

Table 2.1: All Meeting Minutes

2.4 Risk Management Plan

2.4.1 Risk Identification

No	Risk Name	Quantification	Description	Response
1	Project Manager lacks of project management skill	High	- PM don't have management skill - Don't distribute works reasonably	- List and distribute work clearly. - Create schedule to manage deadline, person, task etc. - Improve management skills
2	Unclear requirements	High	- Team members do not understand requirements clearly.	- Review a prototype.

			- Requirements are wrong.	
3	Poor experience about some domain knowledge	High	- Project need some skill to complete task - All team members do not have experience about Arduino	- List tasks and check continuously. - Evaluate quality and progress weekly. - Training some new technique like: Android, Arduino
4	The project is not on time.	High	- Underestimate time of project. - The scope of project is too big for team. - Spend too much time to research and study but don't start coding.	- Keep track with capstone's deadline. - Ask supervisor for advice to define suitable scope. - Put high priority on main functions. - Start coding as early as possible. - Work together for tracking schedule.
5	Conflict between team members	Medium	- Opinion disagreements - Some team members don't get along with others.	- Organize some team buildings to bond members together. - Talk with each member to find the solution.
6	Data lost	High	- Lost data because do not backup the data. - Someone merge code make lost another one code.	- Define a schedule to back up project's data - Restore project's data. - Using SVN repository to back- up data.
7	Equipment Supplier	Medium	- Limited number of supplier, transaction complexity	- Find replaced supplier Contact with supplier at the beginning of project.

8	Complex system integration	High	- Modules integration not matches. - Application on PC and microcontroller does not active	- Check the activity of individual modules. - Deploy person with prior experience with the domain. - Integration test - System test
9	Scalability problems	Medium	- The system isn't stable, difficult to installation	- Release beta version to check stability and debug error

Table 2.2: Risk identification

2.4.2 Risk control

- Close settled risks and identify new risks.
- Help control risks by performing analyses of various mitigation options.
- Once a risk has been identified and a plan put in place to manage it.
- Determining Mitigation Plans: recognize risks, encourage risk taking.
- Include risk monitoring as part of the program review and manage continuously.
- Revisit risk analysis as plans and actions are successfully completed.
- Review and track risk mitigation actions for progress.

2.5 Configuration Management

2.5.1 Introduction

The purpose of this document is to identify an describe Configuration management (CM) process implementing in the project

2.5.2 Roles and Responsibilities

Refer to 2.2.2 section

2.5.3 Configuration management process

2.5.3.1 Project Infrastructure

Tool	Purpose	Note
Eclipse	For development	*https://www.eclipse.org/downloads*
TortoiseSVN	For source code control	
Code Google	For document and source storage	http://code.google.com

2.5.3.2 Directory structure

Main Folder	Sub Folder	Purpose
Include_module_finnal_embeded		To Store final product
Software		Store software that is used by member.
Report		Store Reports
Module	bluetooth	Store Bluetooth's source code
	display_inf_sensor	Store Sensor's source code
	learn_and_display_for_member	Store Source code
	mobile_display	Store App's source code

Table 2.3: Directory structure

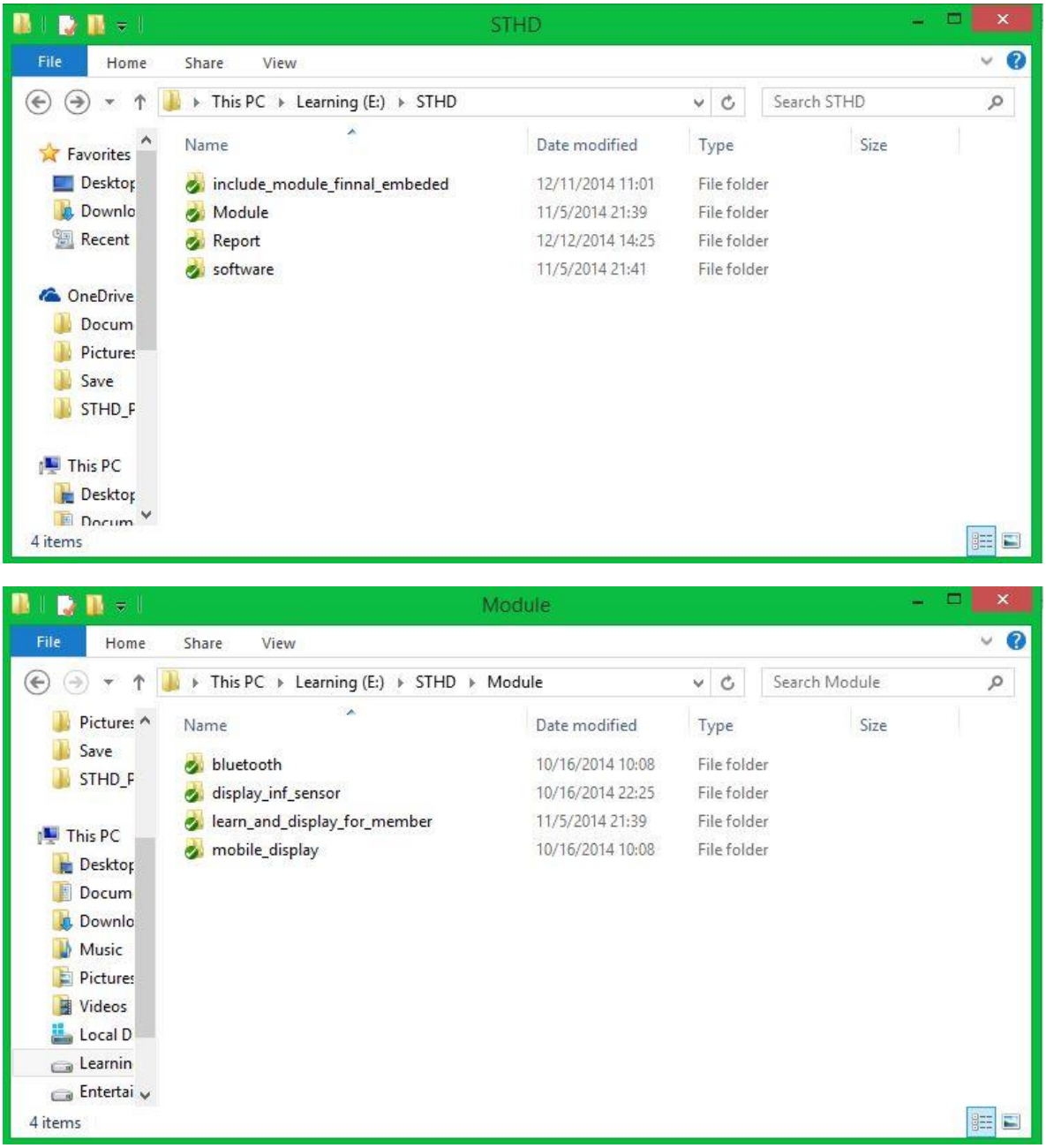


Figure 2.4: Directory structure

2.5.3.3 Baseline Procedure

2.5.3.3.1 For Document

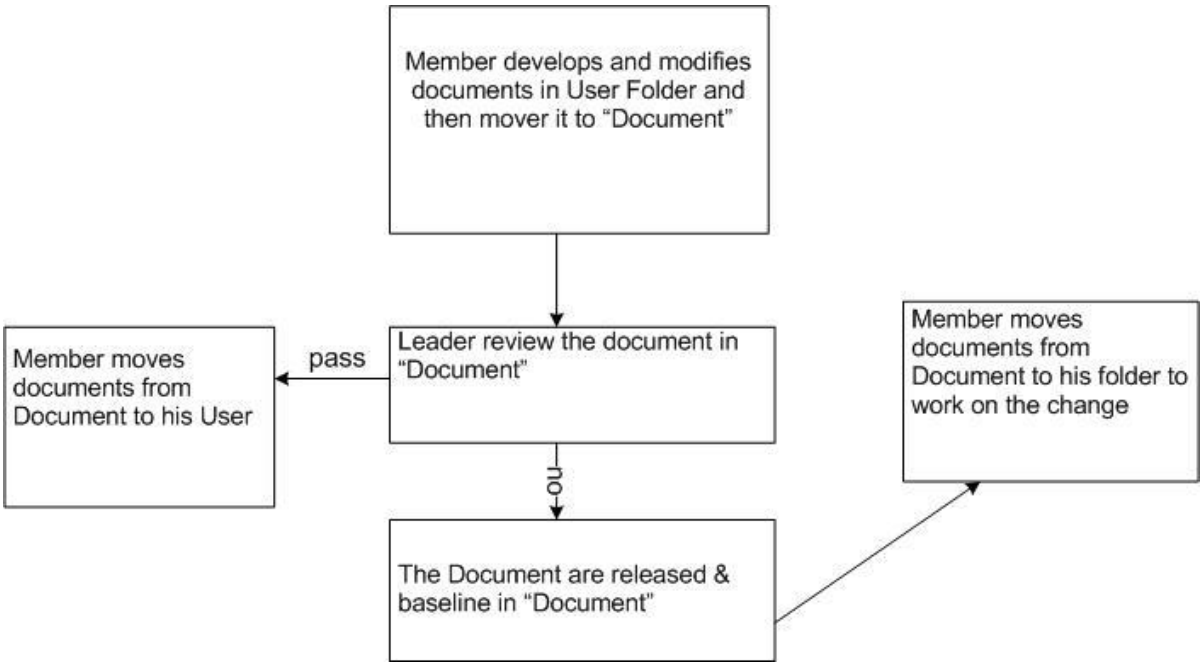

Member develops and modifies documents in User Folder and then mover it to "Document"
Leader review the document in "Document"
pass
Member moves documents from Document to his User
no
The Document are released & baseline in "Document"
Member moves documents from Document to his folder to work on the change

Figure 2.5: For Document

2.5.3.3.2 For Source Code

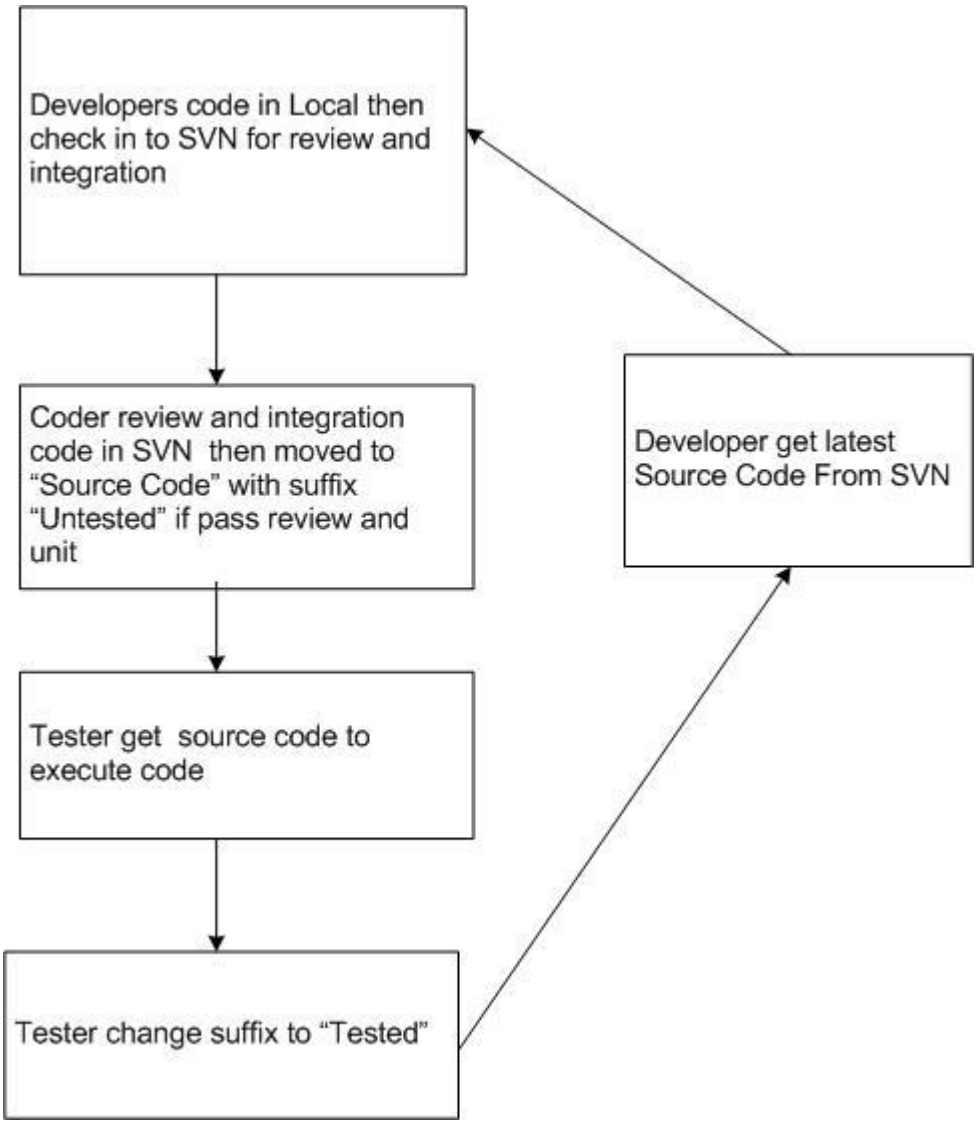

Developers code in Local then check in to SVN for review and integration
Coder review and integration code in SVN then moved to "Source Code" with suffix "Untested" if pass review and unit
Tester get source code to execute code
Tester change suffix to "Tested"
Developer get latest Source Code From SVN

Figure 2.6: For Source Code

2.6 Coding Convention

Follow coding convention. See the reference for detail:

- Android: *https://source.android.com/source/code-style.html*
- Arduino: *http://arduino.cc/en/Reference/StyleGuide*

2.7 Communication Plan

Meeting with supervisor: When we are difficult to complete task or need experience about domain knowledge and everything that can't be solved by ourselves, we will ask our supervisor for advises.

Meeting: Team member meeting on Monday, Tuesday and Thursday. During the meeting, members will work together, discuss problems, share the technical experience and have been in a work on week. Team leader list tasks which will be must to complete on next week.

Communication channel: Our main communication channels are email, Facebook and Skype. Its help us to communicate, share the difficulties encountered easily or help other members complete task. Sometimes we have an emergency video call by Skype.

CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATIONS (SRS)

3.1 User Requirement Specification

- Lower cost than current system.
- High performance and quiet operation.
- Friendly user interface and easy to use.
- When operating, system must be able to:
    - Stably and accurate
    - Safe and reliability
    - Power management: minimize heat production
    - Back up the system set up.
- Being able to send data between mobile android and Arduino
- User able to view value of temperature and humidity in wooden box on android mobile.
- User able to set value of temperature and humidity from android mobile.

3.2 System Requirement Specification (Specific Requirements)

3.2.1 External Interface Requirements

3.2.1.1 User Interfaces

User interface is very important to communication between user and system. The interface should present and acquire information. It must be satisfied the requirements of all users:

- The interface should be clearly, friendly and easily to use.
- The interface should have a main screen and a paired screen, it include the following functions:
    - Display the information about current temperature/humidity.

- Set up the temperature/humidity.
- Clear and Send command.
- Turn ON/OFF Bluetooth
- Search button is used to list all devices that can be connected to the system.
- Disconnect form the system.

- When user tap on "Set" button to set up the temperature and humidity. Temp/Hum value that is set will be sent to the system via Bluetooth to control temperature and humidity change.
  - A Seekbar is used to set temperature/humidity.
  - A Textview display temperature/humidity which user set in seekbar.

3.2.1.2 Hardware Interfaces

- Arduino kit:
  - Arduino Uno use 5V DC.
- Android devices:
  - Android devices using Android 4.1 or higher.
  - The Bluetooth of devices is 2.4GHZ.
  - The devices have installed *Bluetooth SPP* to control Bluetooth communicate.
- HC-05 Bluetooth module:
  - The frequency of module is 2.4GHZ
- DHT22 temperature-humidity sensor:
  - 3 to 5V power and I/O.
  - Good for 0-100% humidity readings with 2-5% accuracy
  - Good for -40 to 80°C temperature readings ±0.5°C accuracy
- USB TO COM PL2303:
  - Configure Bluetooth module use Hercules Setup Utility terminal on PC.
- PC:
  - The PC have installed driver for USB TO COM PL2303.
  - The have installed Hercules Setup Utility.
  - Support USB 2.0 port or higher.
  - The PC has installed Arduino IDE.

- Window 7/8/8.1

3.2.1.3 Software Interfaces

- Application must be code in Android programming language.
- Application for Android is compatible with the devices using Android 4.1 or higher.

3.2.1.4 Communications Protocol

- Load programs onto Arduino via UART port use cable USB printer.
- Sent AT command from PC to Bluetooth module to configure it via UART port user cable USB TO COM PL2303.
- Arduino and Bluetooth use UART direct connection (RX Arduino – TX Bluetooth; TX Arduino – RX Bluetooth).
- Bluetooth module and Android devices use Bluetooth 2.4GHZ connection

3.2.2 Functional Requirements

3.2.2.1 Use Case Diagram

Application on Android

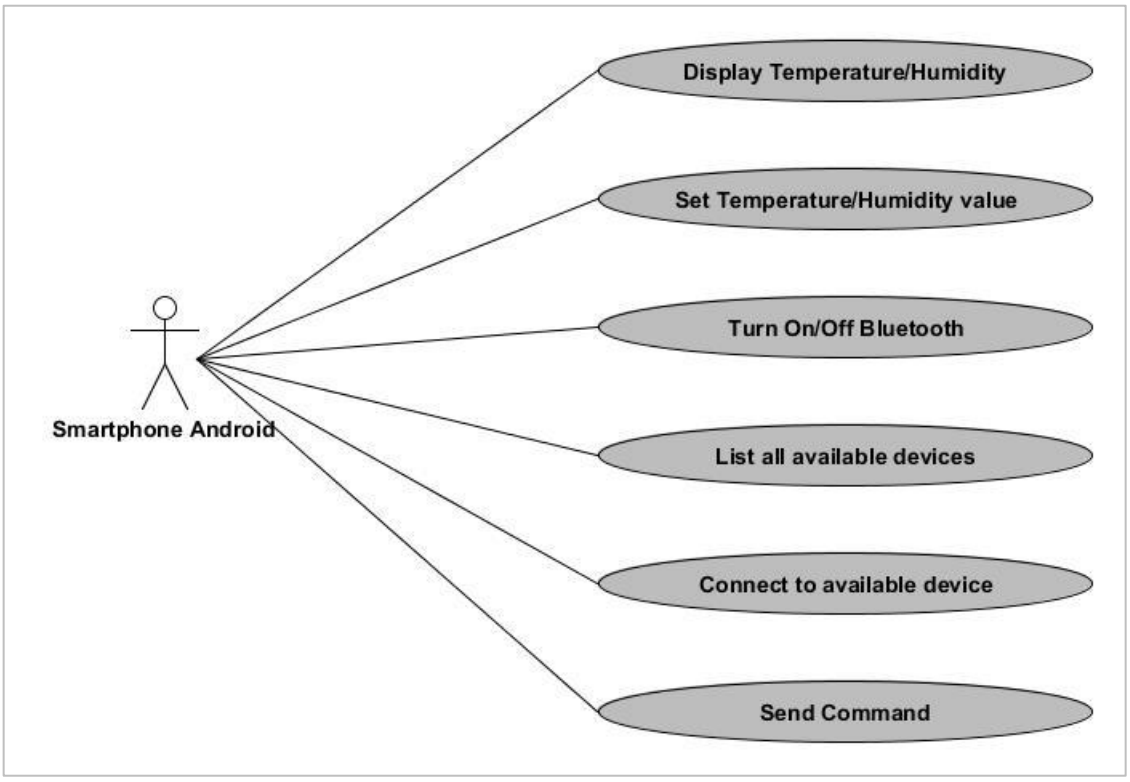


Figure 3.1: Use Case on Smartphone

Controller Box

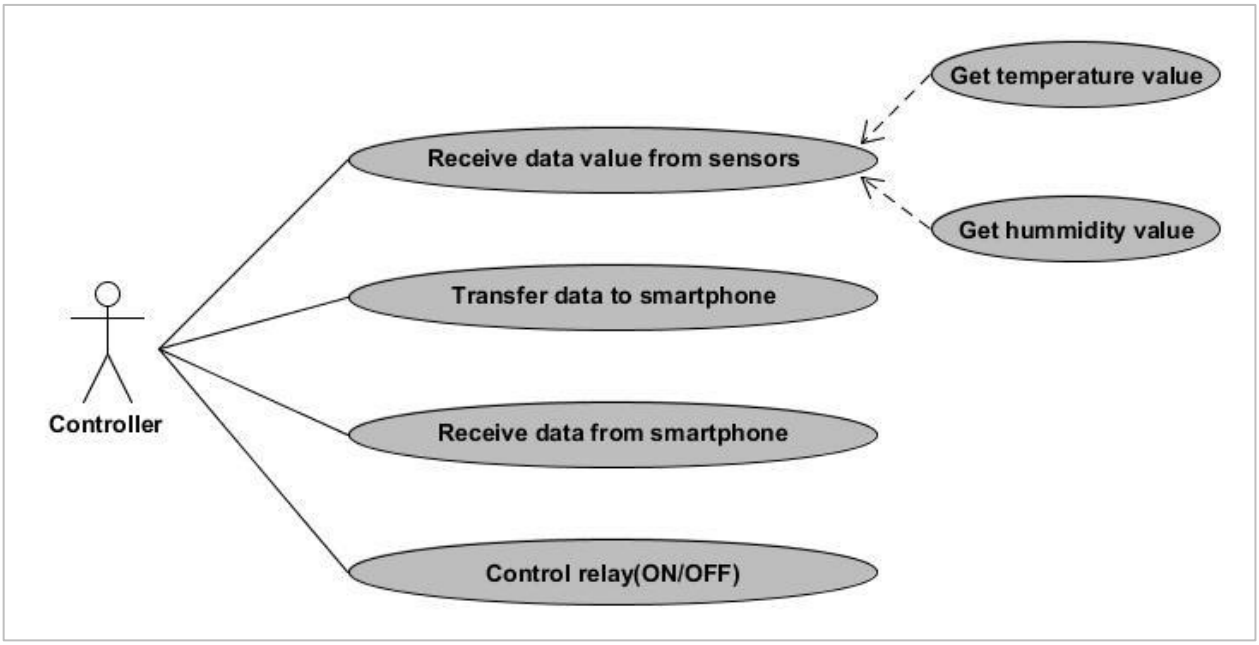


Figure 3.2: Use Case on Controller Box

3.2.2.2 Use Case Specification

Function 01	**Display temperature and humidity**
Description	User can view current temperature and humidity value from the environmental chamber on smartphone's screen.
Input	
Source	Temperature and humidity value from Arduino.
Output	The temperature and humidity value is displayed on application interface.
Action	- When user start to use application, the temperature and humidity value will be got from Arduino in Bluetooth. - Application will display information on main screen.
Pre-condition	- Software application is running. - System is running.
Post-condition	

Function 02	**Set value of temperature/humidity up**
Description	User can use application on smartphone Android to set up temperature/humidity value. It will be send Arduino to control relay.
Input	[Value of temperature] [Value of humidity]
Source	Number
Output	Temperature/Humidity value is transferred to Arduino
Action	- User tap [Set] button on temperature box or humidity box to set it up. - After tapping [Set] button, A dialog will be displayed on screen. - On dialog, Seekbar is used to set temperature/humidity up. It will be displayed on Textview. - After setting up, user can choose [Cancel] button or [OK] button. - If you choose [OK] button, data will be transferred to Arduino.
Pre-condition	- Software application is running. - System is running.
Post-condition	

Function 03	**Turn On/Off Bluetooth**
Description	User can turn on or turn off Bluetooth with Android device
Input	
Source	
Output	Bluetooth program is disable/enable.
Action	- When user clicks the Turn On/Off button, the Bluetooth program is enable/disable.
Pre-condition	- Software application is running. - System is running.
Post-condition	

Function 04	**List all available device**
Description	User can see available devices list that can be connected to the system
Input	
Source	
Output	
Action	- When user clicks the Search button, all devices that can be connected to the system is displayed on screen.
Pre-condition	- Software application is running. - System is running.
Post-condition	

Function 05	**Connect to available device**
Description	User can connect the Android smartphone with the system
Input	
Source	
Output	
Action	- When user clicks the Search button, all devices that can be connected to the system is displayed on screen. - User can choose one of the devices to connect to the system.
Pre-condition	- Software application is running. - System is running.
Post-condition	

Function 06	**Send Command**
Description	Application allows user to send command to the Arduino and Bluetooth module to configure its.
Input	String that is entered by users.
Source	

Output	Command that configure the setting.
Action	- User enter the command in the textbox - Tap on “Send” button to send Command to the devices.
Pre-condition	- Software application is running. - System is running.
Post-condition	

Function 07	**Get temperature value**
Description	Sensor collects information of temperature in chamber.
Input	
Source	Temperature in environmental chamber.
Output	
Action	- Sensor collects information of temperature in chamber. - Sensor uses a single-wire to send data to Arduino.
Pre-condition	- System is running.
Post-condition	Sensor collected information of temperature

Function 08	**Get humidity value**
Description	Sensor collects information of humidity in chamber.
Input	
Source	Humidity in environmental chamber.
Output	
Action	- Sensor collects information of humidity in chamber. - Sensor uses a single-wire to send data to Arduino.
Pre-condition	- System is running.
Post-condition	- Sensor collected information of humidity

Function 09	**Receive data from sensor**
Description	System allows Arduino Uno to get temperature/humidity information from DHT 22.
Input	
Source	Data that sensor collected
Output	
Action	- Arduino Uno is connected via DHT22 in a single-wire. - Arduino Uno received data from sensor.
Pre-condition	- System is running. - Arduino and sensor are connected.
Post-condition	- Arduino Uno received data from sensor.

Function 10	**Transfer data to smartphone**
Description	Arduino receives data from sensor, analyzes data and controls the system. It transfers this information to smartphone in Bluetooth.
Input	
Source	Data that Arduino received or analyzed.
Output	
Action	- Arduino is connected via Smartphone in Bluetooth - Arduino connects Smartphone and transfer data that it collected and analyzed.
Pre-condition	- Application is running - System is running.
Post-condition	- Data is transferred to smartphone.

Function 11	**Receive data from smartphone**
Description	Application allows user to set temperature/humidity up to control the balance of temperature on system. Arduino received this data from smartphone to control the system.
Input	
Source	Data that user set up on smartphone.
Output	
Action	- Arduino is connected via Smartphone in Bluetooth. - Arduino connects Smartphone and receives data.
Pre-condition	- Application is running - System is running.
Post-condition	- Arduino Uno received data from smartphone.

Function 12	**Control relay(ON/OFF)**
Description	When Arduino transferred and received data, it analyzes and compared this data to control relays into ON or OFF. System will be automatic.
Input	
Source	Data that Arduino transferred and received
Output	
Action	- Arduino receive data from sensor, it transfer this data to smartphone. - Smartphone will display information of temperature/humidity on screen. - User can view the display and set value of temperature/humidity. - Then, Arduino receive data from smart phone. - Arduino controls relays station.
Pre-condition	- Application is running - System is running.
Post-condition	- Relays can be changed automatically from ON to OFF or change

	from OFF to ON.

Table 3.1: Use Case Specification

3.2.3 Non-Functional Requirements

3.2.3.1 Usability

- The system's interface must be clear and simple.
- The system is friendly and self-explanatory.
- More efficient to use-takes less time to accomplish a particular task.
- Easier to learn-operation can be learned by observing the object.
- More satisfying to use.

3.2.3.2 Reliability

- The system is available 98% up-time to the user.
- All functions operate exactly to test hardware device.
- On smartphone, application must operate, display information and control exactly.

3.2.3.3 Security

- When user connects to system in Bluetooth, user must type password.

3.2.3.4 Maintainability

- The system can be updated easily when demand of user change or improve.
- Easy to replace fail devices.

3.2.3.5 Delivery

- The documentation must be delivered in December 17th 2014
- The system can be implemented after the document is delivered

CHAPTER 4: IMPLEMENTATION

4.1 Proposed System Architecture

4.1.1 Flow Chart

1. Control Relay ON/OFF

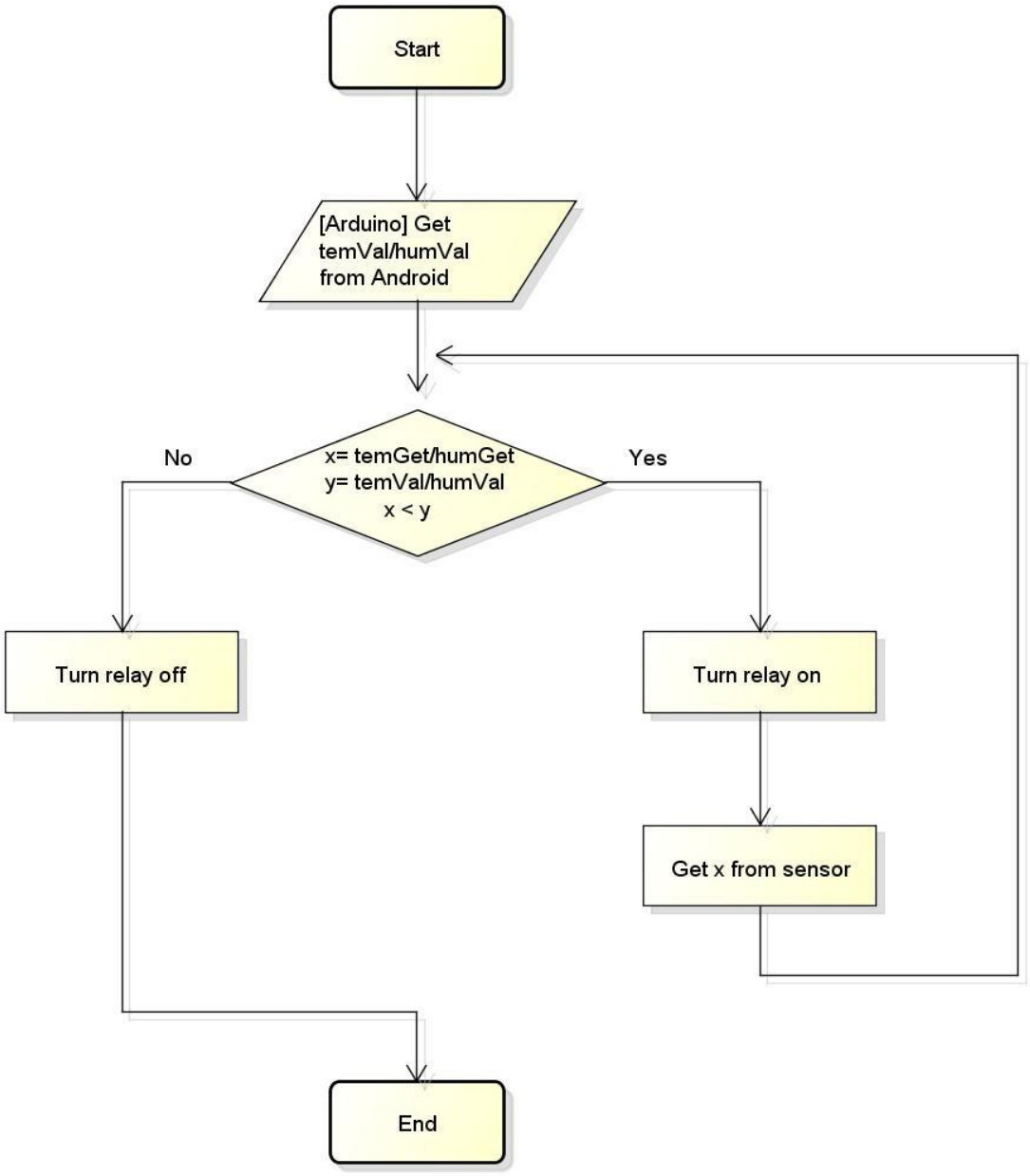


Figure 4.1: Control Relay ON/OFF

2. Display temperature and humidity

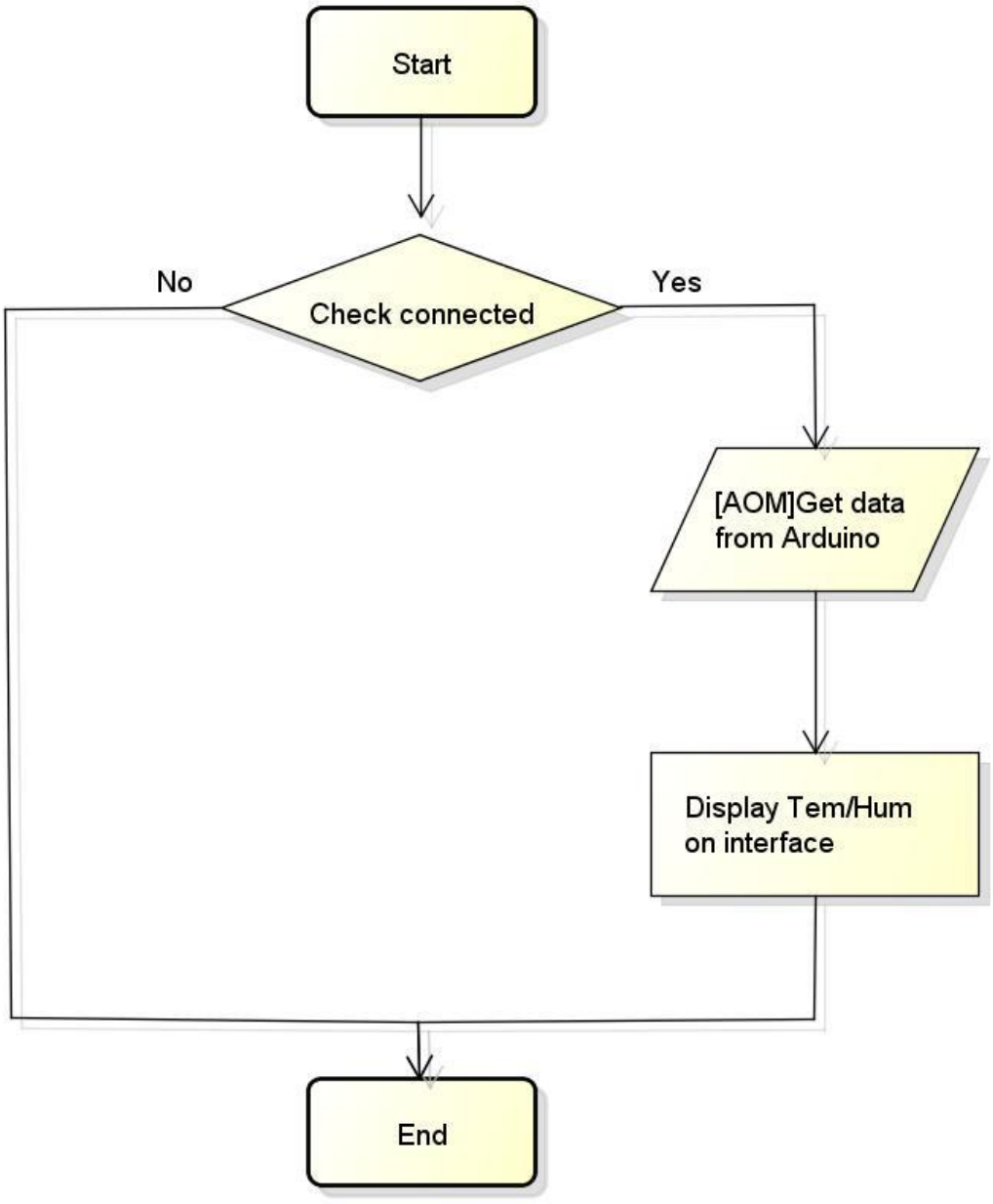


Figure 4.2: Display temperature and humidity

3. Receive data

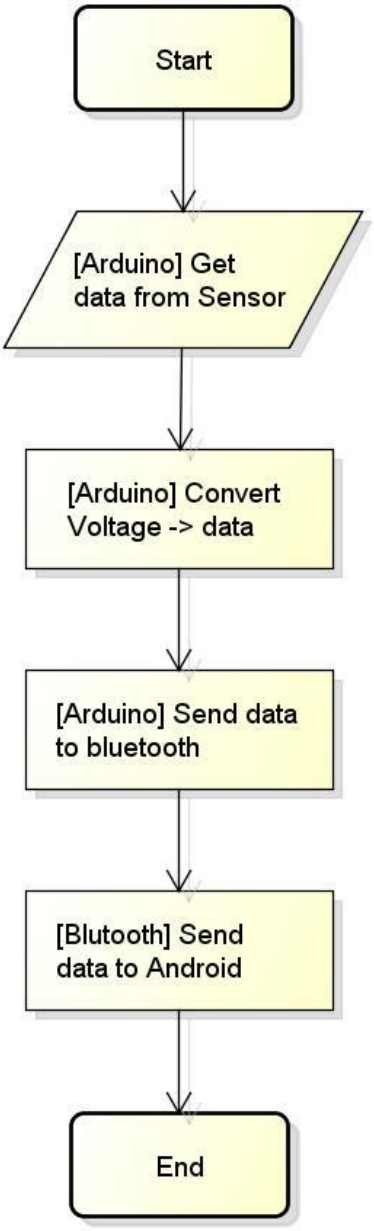


Figure 4.3: Receive data

4. Set value

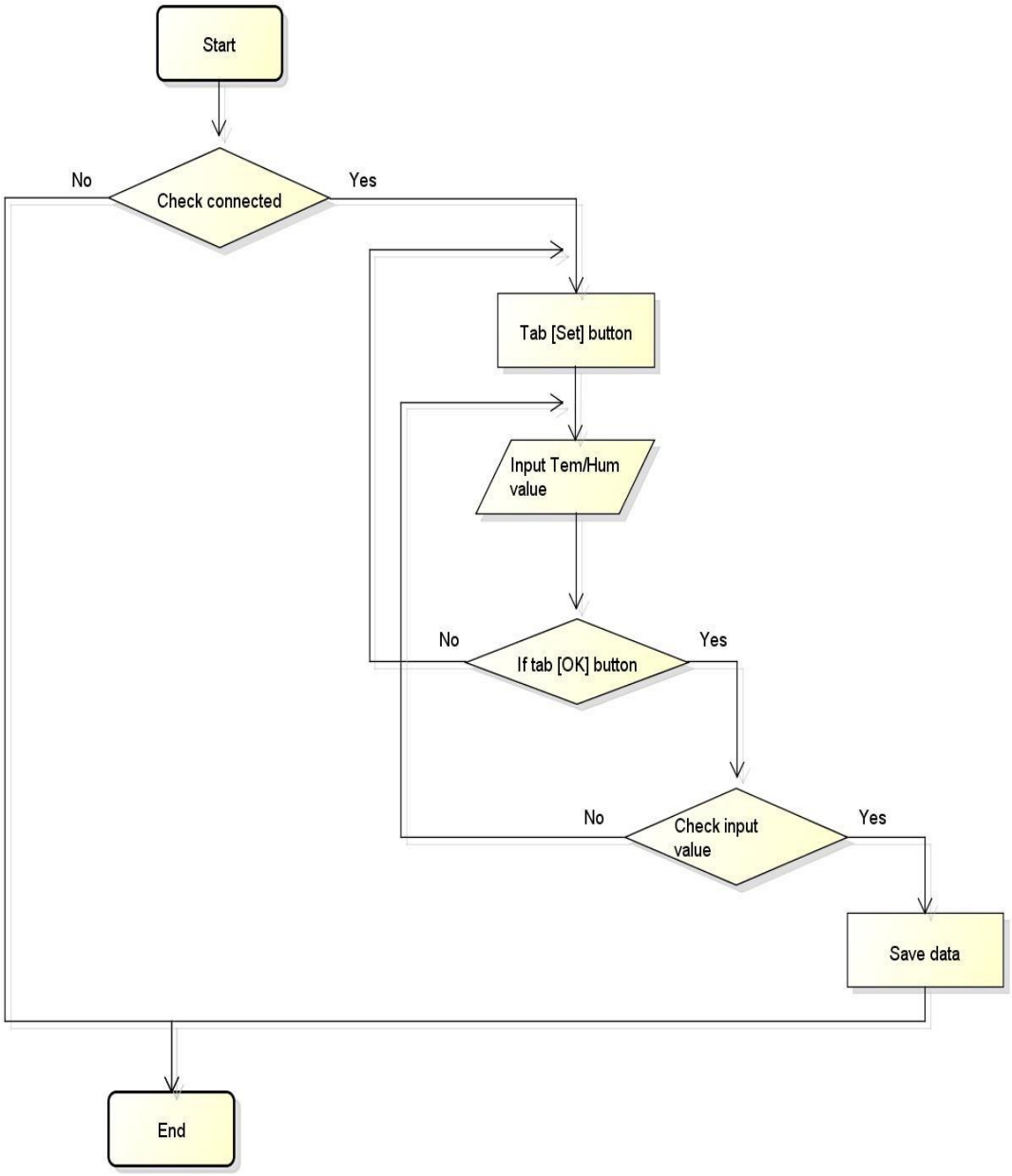


Figure 4.4: Set value

4.2 Analysis and Selection of Tool, Devices

4.2.1 Software

4.2.1.1 Arduino Development

1. Arduino IDE

- The open-source Arduino environment makes it easy to write code and upload it to the I/O board.
- Can runs on Windows, Mac OS X, and Linux.
- The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

2. Hercules Setup utility 3.2.8

- Support communication between the remote device's controls to the PC.
- Hercules is great utility working with the serial and Ethernet devices & interfaces.
- Easy to use.
- Useful with:
    - Serial port terminal (COM5 and higher support).
    - TCP/IP Client terminal (similar to "Telnet").
    - TCP/IP Server "terminal".
    - UDP "terminal".

4.2.1.2 Design and coding tool

1. Eclipse

- Easy to use.
- Auto complete in methods: As you write your code, you will be presented with a list of possible choices.
- As you expand a source file, you will see a list of its classes and a list of methods in each class and so on. This can help you look up a method

quickly if needed. You can also use the outline pane as you are writing code to have a quick list of accessible methods

- To add libraries to your project that you will use such as provided jar files
- Provides reconciled set of possibilities for most of the platforms.
- Apart from Java, Android, it supports other languages.
- It has a free and an open source with a full support as well
- Many flexible tools and frameworks for building various types of applications.
- It is really extendable and configurable.
- Industrial level of development.

2. Astah

- Very easy to use.
- Very lightweight, sufficiently complete, and great for modeling.
- Astah give you power and many futures to draw standard diagram.
- Easy to convert between diagrams: convert Flowchart models to UML models, convert ER models to DFP models…
- Exporting to JPEG, PNG, EMF, SVG files.

3. Proteus

- Basic and easy tools to design circuit.
- Full features schematic capture.
- Work with popular compliers and assembler.
- Interactive peripheral models for display, keypads…
- It is integrated with real time simulation of the electronic circuit and test whether your designed circuit is working properly or not
- In this software you can test your test code by loading the hex file in the circuit and test the circuit proper functioning.
- Proteus PCB design products include an integrated shape based auto router and a basic SPICE simulation capability as standard.

4. Google sketup

- SketchUp is a 3D modeling software tool that combines a tool-set with an intelligent drawing system.
- The system interfaces with smart graphics cursor allows users to draw three-dimensional rendering in two-dimensional space of the screen.
- It is free
- It is great for working fast and loose in 3D
- Apply colors and textures
- See inside your models
- Export TIFF, JPEG and PNG

4.2.2 Hardware

4.2.2.1 Arduino Uno

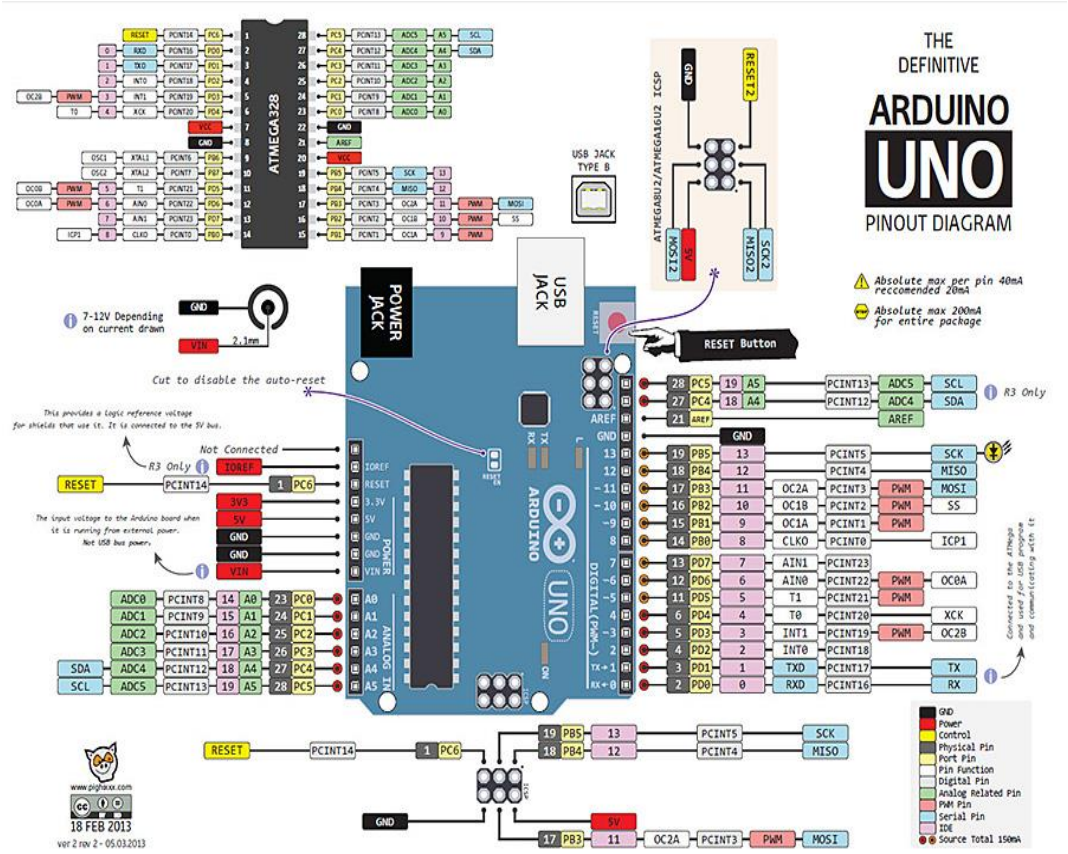


Figure 4.6: Arduino Uno Schematic

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- GND. Ground pins.

- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), anddigitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

4.2.2.2 HC-05 Serial Port Bluetooth Module

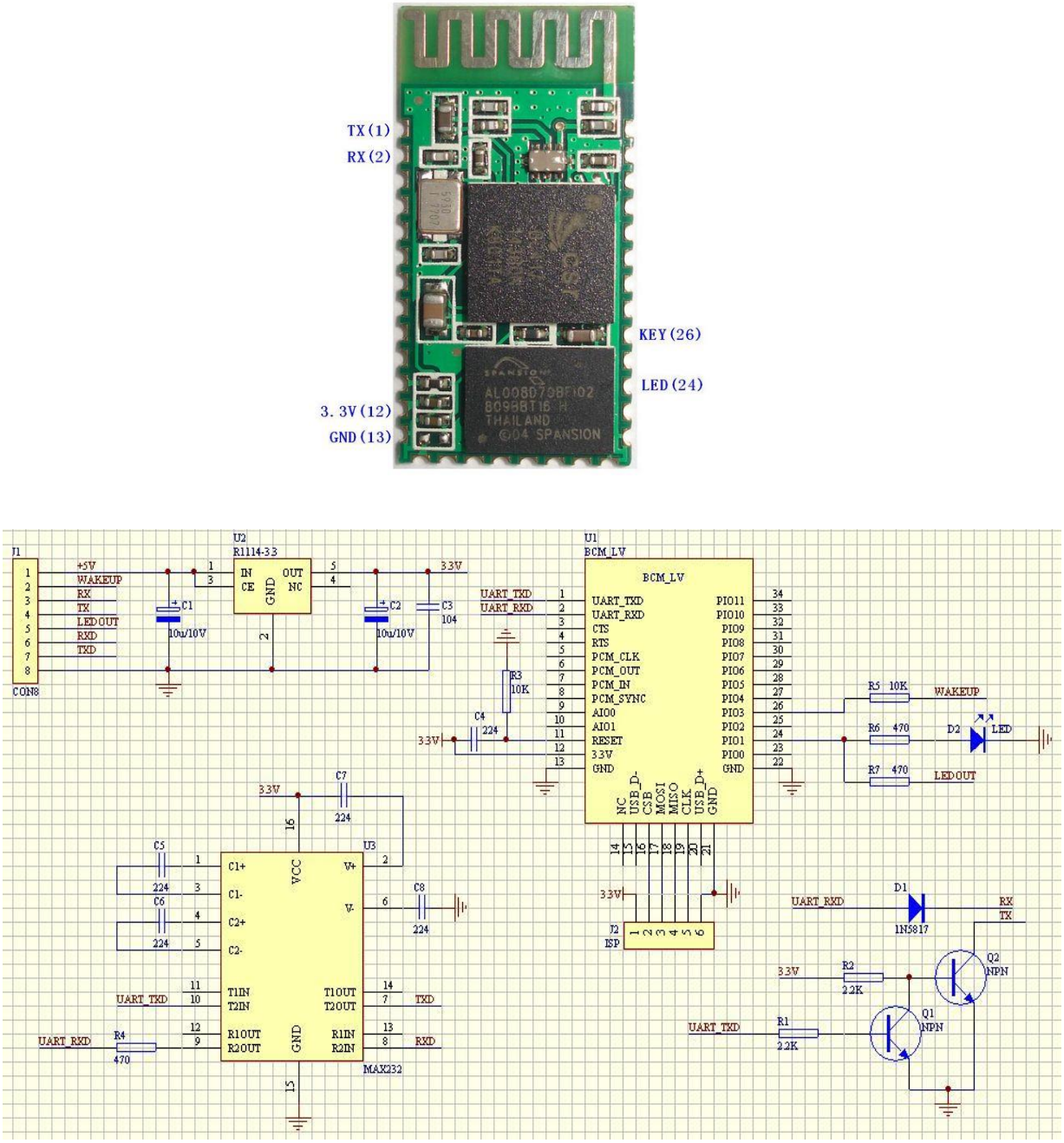


Figure 4.7: HC-05 Serial Port Bluetooth Module

PINs description

PIN1	• UART_TXD , TTL/CMOS level, UART Data output
PIN2	• UART_RXD, TTL/COMS level, s UART Data input
PIN11	• RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	• VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	• GND
PIN22	• GND
PIN24	• LED, working mode indicator Slave device: Before paired, this PIN outputs the period of 102ms square wave. After paired, this PIN outputs high level. Master device: On the condition of having no memory of pairing with a slave device, this PIN outputs the period of 110ms square wave. On the condition of having the memory of pairing with a slave device, this PIN outputs the period of 750ms square wave. After paired, this PIN outputs high level
PIN26	• For master device, this PIN is used for emptying information about pairing. After emptying, master device will search slaver randomly, then remember the address of the new got slave device. In the next power on, master device will only search this address.

4.2.2.3 DHT22 temperature-humidity sensor

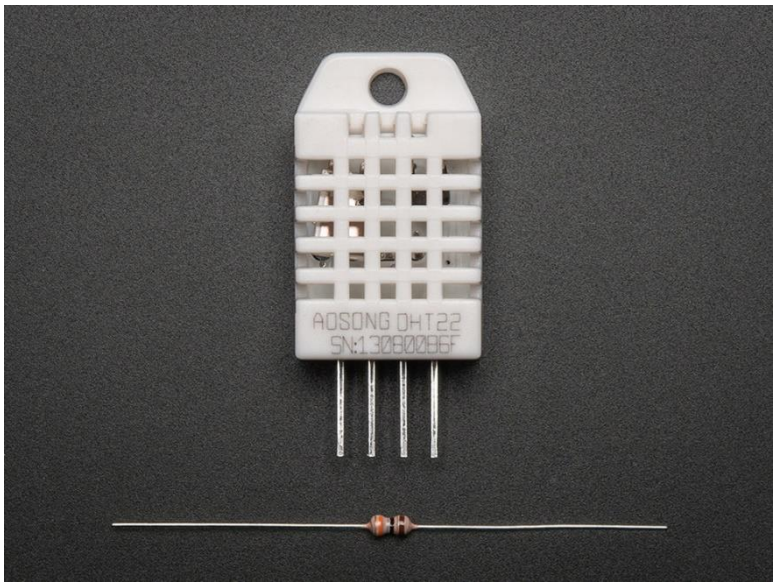


Figure 4.8: DHT22 temperature-humidity sensor

Pin sequence number: 1 2 3 4(from left to right direction)

Pin	Function
1	VDD----power supply
2	DATA--signal
3	NULL
4	GND

4.2.2.4 4-Channel 5V Relay Module

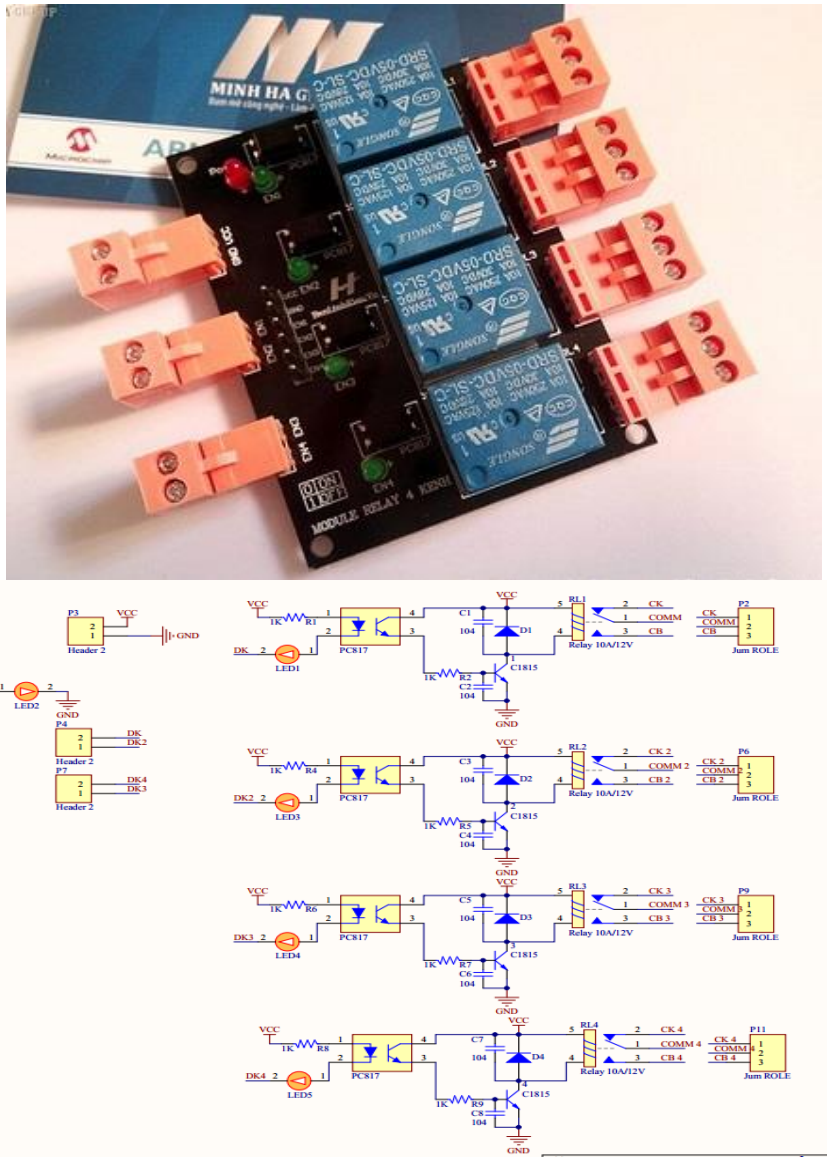


Figure 4.9: 4-Channel 5V Relay Module

Features

- 4 mechanical relays with status indicator LED
- Both “NC” and “NO” ports for each relay

SSR Output (each channel)

- Load voltage range: 75 to 264VAC (50/60Hz)
- Load current: 0.1 to 2 AMP

Input control signal voltage

- 0V - 0.5V Low stage (SSR is OFF)
- 0.5V – 2.5V (unknown state)
- 2.5V - 20V High state (SSR is ON)

4.3 Circuit Design

4.3.1 Circuit for Fox-301A System

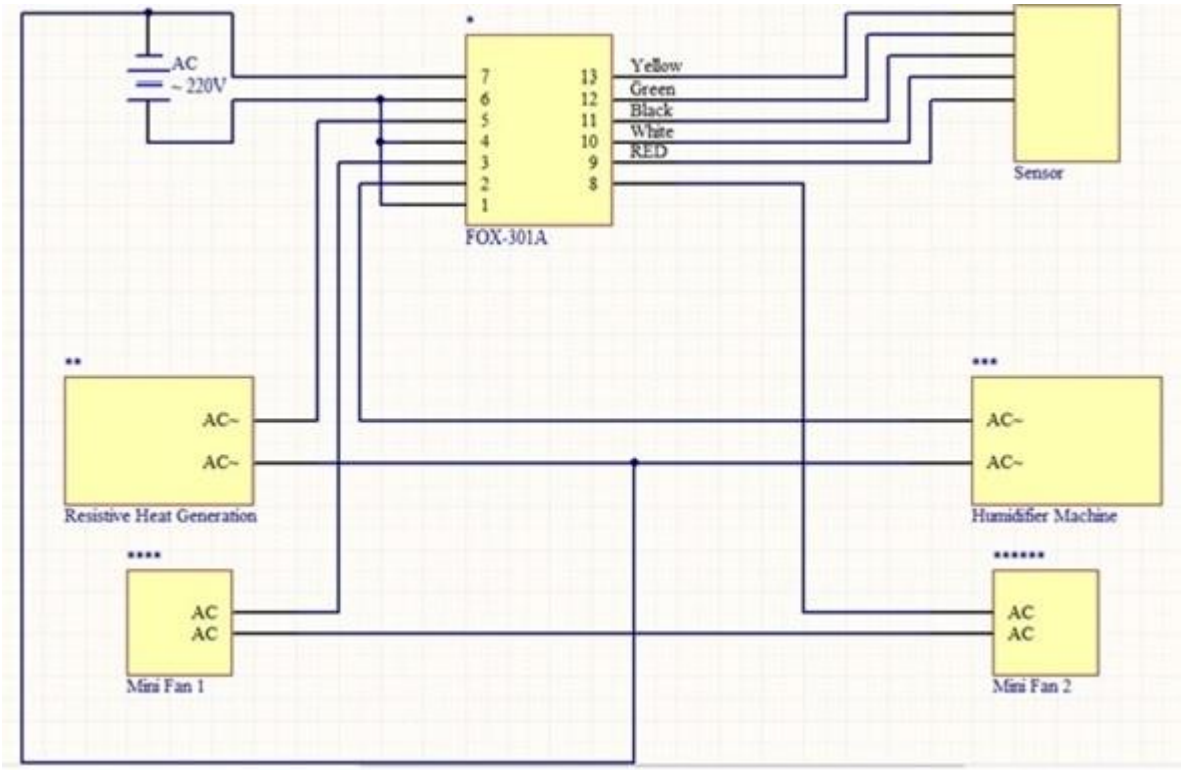


Figure 4.10: Fox-301A System Circuit

Summary

- FOX-301A is a temperature and humidity controller.
- FOX-301A includes a control and displays box, a temperature and humidity sensor.
- Packaged by the manufacturer.
- Automatic operation.

How to installing the components together

- Use 220V power.
- 5^{th} and 3^{rd} pins are the outputs of 4^{th} pins. 2^{nd} and 8^{th} pins are the outputs of 1^{st} pins.
- Sensor has been built according to the color of the wire.
- 7^{th} and 6^{th} pins of the FOX-301A are connected to the 220V power.

- One pole of the power AC~220V connect to Humidifier Machine, Resistive Heat Generation, Mini Fan1 and Mini Fan2. The remaining pole of the power AC~220V connect to 4th and 1st pins of the FOX-301A.
- 5th pins of the FOX-301A connect to the remaining pole of the Humidifier Machine. 3rd pins of the FOX-301A connect to the remaining pole of the Mini Fan1.
- 2nd pins of the FOX-301A connect to the remaining pole of the Resistive Heat Generation. 8th pins of the FOX-301A connect to the remaining pole of the Mini Fan2.

Operating principle

- Values of temperature and humidity, which we desire to be set by the buttons on the control box.
- Sensor measured values of temperature and humidity in the environment chamber and passed to control box. Control box will compare the value received from Sensor and desired value, then make a decision to turn on or off switches.
- If the temperature needs to increase, switch between 4th and 5th pins will turn on. If the temperature needs to decrease, switch between 4th and 3rd pins will turn on.
- If the humidity needs to increase, switch between 1st and 2nd pins will turn on. If the humidity needs to decrease, switch between 1st and 8th pins will turn on.

4.3.2 Circuit for PID System

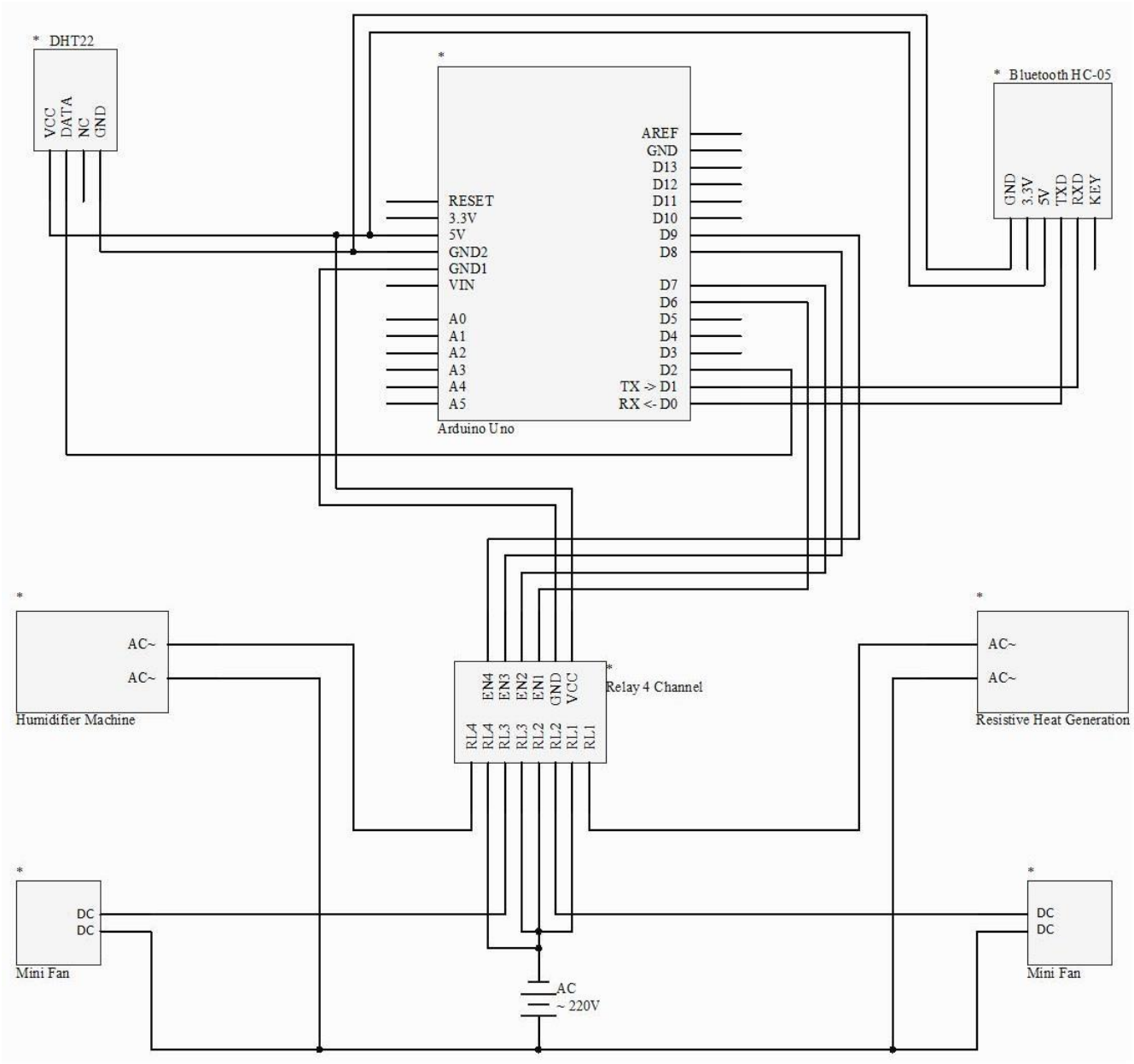


Figure 4.11: STHD System Circuit

- Arduino Uno: is a central signal processing and control device. Bluetooth HC05, DHT22 Sensor, Relay 4Channel are supplied a power source 5V by Arduino Uno
- Module Bluetooth HC05: TXD and RXD pins of the HC05 connect in turn to the RXD and TXD pins of the Arduino Uno (TX $^{\text{Bluetooth HC05}}$ with RXD0$^{\text{Arduino Uno}}$, RXD

$^{\text{Bluetooth HC05}}$ with TXD1 $^{\text{Arduino Uno}}$), these pins will help Arduino Uno and Bluetooth HC05 transmit and receive signals together. GND and 5V pins of the HC05 connect in turn to the GND and 5V pins of the Arduino Uno.

- DHT22 Sensor: GND and VCC pins of the DHT22 connect in turn to the GND and 5V pins of the Arduino Uno. DATA pins of the DHT22 connect to D2 pins of the Arduino Uno
- Relay 4 channels: EN1, EN2, EN3 and EN4 pins of the Relay connect in turn to D6, D7, D8 and D9 pins of the Arduino Uno. GND and VCC of the Relay connect to GND and 5V pins of the Arduino Uno
- Humidifier Machine, Resistive Heat Generation, and Mini Fan: All these devices are similar connect and use the power 220V. One pole of the power AC~220V connect to Humidifier Machine, Resistive Heat Generation and Mini Fan. The remaining pole of the power AC~220V connect to inputs of the switch RL1, RL2, RL3 and RL4. Output of this switch will connect to the Humidifier Machine, Resistive Heat Generation, and Mini Fan.

Operating principle

- Android Bluetooth Device transmits for Bluetooth HC05 a value of the temperature and humidity then this value will be transmitted to Arduino Uno.
- DHT22 Sensor measured values of temperature and humidity in the environment and passed to Arduino Uno.
- Arduino Uno will compare the value received from Bluetooth HC05 and DHT22 Sensor, and then make a decision to turn on or off switches (RL1, RL2, RL3, and RL4).

4.4 Algorithm and Solution Analysis

4.4.1 Introduction PID Controller

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a

desired setpoint. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element.

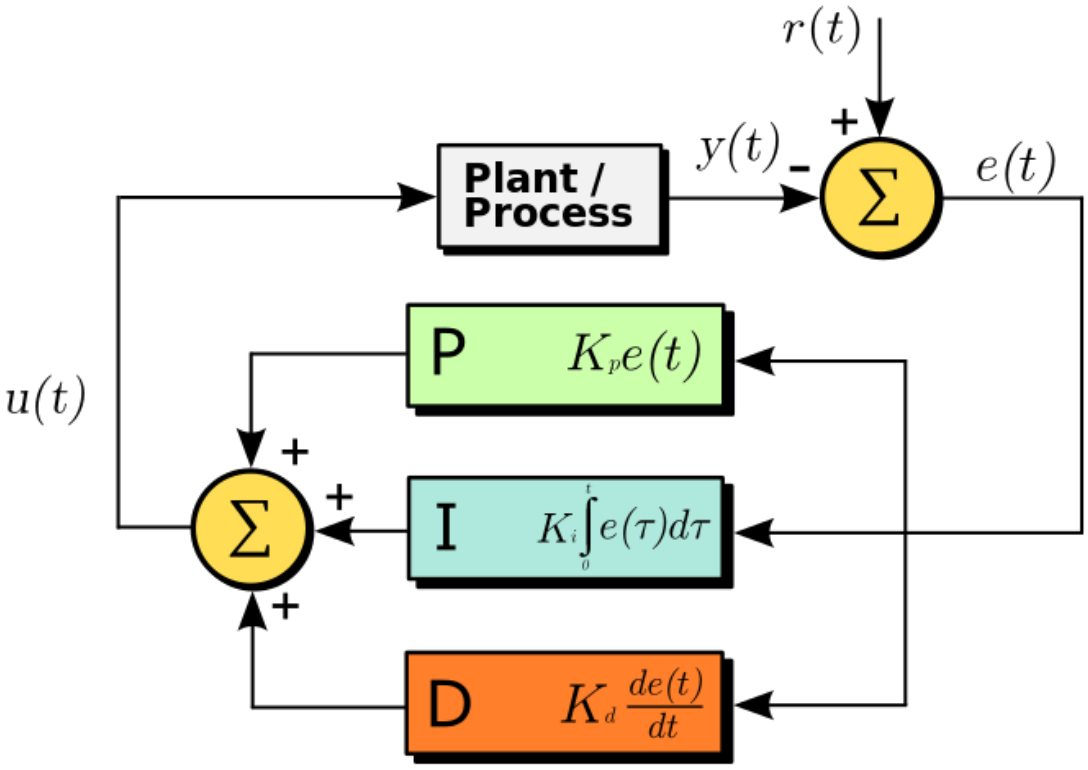


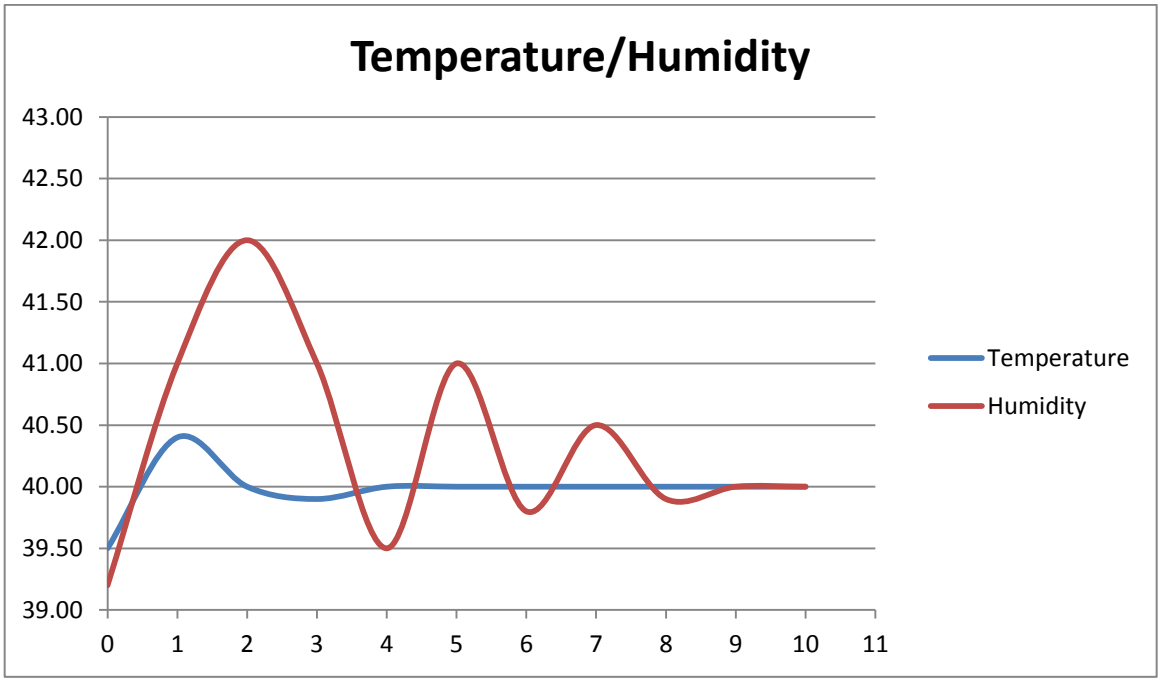
Figure 4.12: A block diagram of a PID controller in a feedback loop

In the absence of knowledge of the underlying process, a PID controller has historically been considered to be the best controller. By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots

the setpoint, and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

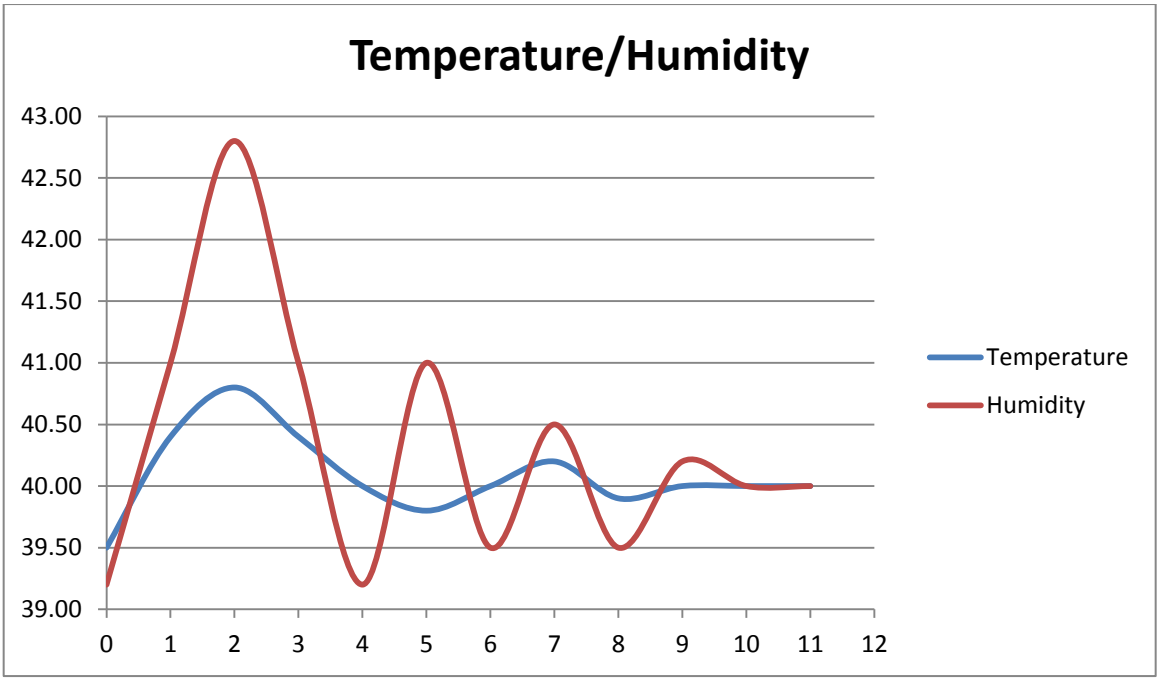
4.4.2 Using Fox-301A on System

Fox-301A is used to develop a prototype**.** On the system that uses Fox-301A to control, we measure and record temperature, humidity on environment chamber. We test the temperature, ranging from 20^{o}C to 40^{o}C and humidity, ranging from 20^{o}C to 40^{o}C .After testing, we have the following chart.



4.4.3 Using PID on System

Arduino Uno is used to develop a testing system. On the system that uses PID to control, we measure and record temperature, humidity on environment chamber. We test the temperature, ranging from 20^{o}C to 40^{o}C and humidity, ranging from 20^{o}C to 40^{o}C .After testing continuously, we use PID (2, 5, 1) to configure PID so that PID will be as same as Fox-301A. After testing, we have the following chart.



4.5 Interface Design

4.5.1 Interface for Box

The steel box is used to create the environment for the system. The box has 2 doors: upper door for test chamber and other for device chamber. The window is used to view test chamber. The features of this box:

- The material: steel, iron…
- Size: 60 cm long, 60 cm wide, 80 cm high.
- The size of test chamber: 60 cm long, 60 cm wide, 50 cm high.
- The size of window: 25 cm long, 25 cm wide.

The image of box:

Figure 4.13: The steel box

4.5.2 Application Interface

A. Paired Screen:

This screen allows user to pair Android smartphone with system via Bluetooth. On this screen have three buttons, it includes the following buttons:

- "Search" button: When you click on this button, App will make you turn on the Bluetooth on smartphone. After turning on Bluetooth, lists all available devices that can be connected to system will be displayed. Then, user can choose device to connect to system. When you connected successfully, Paired Screen will be redirected to Main Screen.
- "Turn off" button: This button allow user to turn the Bluetooth off quickly on smart phone.
- "List Paired" button: When devices connected to system successfully, these devices will be saved on list paired. User can click on this button to connect to system. The system will not search available devices. This action is faster to connect than action that user click on "Search" button.

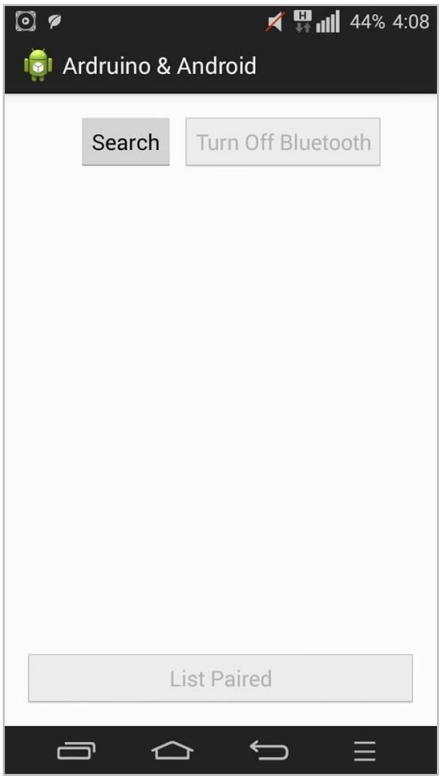


Figure 4.14: Paired Screen

B. Main Screen

Main screen is used to displayed information of temperature/humidity, send command to change the default setting of devices and set Temp/Hum value that is sent to system to control temperature and humidity changes.

- "Disconnect" button: When user click on this button, Android device will be disconnected from the system. Main screen will be redirected to Paired Screen.
- "Set" button: After changing temperature and humidity value, user can set its value for the system.
- Textbox: User can enter string into this textbox.
- "Send" button: Send command that is entered into the textbox by user to the device setting.
- "Clear" button: Clear command that is entered into the textbox by user.

Figure 4.15: Main Screen

CHAPTER 5: SYSTEM IMPLEMENTATION & TEST (SIT)

5.1 Introduction

The overall purpose of testing is to ensure the application and system meet all of its technical, function and business requirements. Product's quality and reliability hold a key post in the software development process. It includes:

- Acceptable response to unusual inputs, outputs, loads and conditions.
- All functions, technical and business requirements as specified and expected.
- Testing process is met expectantly schedule.
- Easy, efficiency of use for tester.
- Ensure that all positive scenarios and negative scenarios are covered.
- Write in simple and easy to understand language.

5.1.1 Test Approach

5.1.1.1 Test type

Function Testing

- Testing of all functions on the STHD system.
- The testing that will be covered for those functions in function test is verification the validation of the behavior according to requirements.
- Test cases will have to be designed to cover all the above and executed. 5.2.1 Features to be tested

User Interface Testing

- GUI test will be performed fully on all screens.
- This test is targeted to test the STHD system including initial position, font, text size, initial button, dialogs.

5.1.1.2 Test levels

- ***Unit test:*** it will be done by the developer and refers to tests that verify the functionality of a specific section of code, usually at the function level. All unit test information (input, output…) will be managed by developer and will not need test cases.
- ***System/Integration test:*** This test stage will be performed by testers. Testers validate all functions and system interfaces on the system that must be met to requirement on SRS. When tester find bug, developer must fix and tester will be recertify bug.
- ***Acceptance test:*** This test stage will be performed by test leader and development leader. It requires the real customers with their own test cases and real environment to test. At last the system is delivered to the user.

5.2 Test plan

5.2.1 Features to be tested

Embedded System

- Test Bluetooth
- Test relay(On/OFF)
- Test sensor

Android Application

- Test user interface
- Test load information
- Test functions
- Test Bluetooth

5.2.2 Test Schedule

No.	Task	Tester	Effort	Start Date	End Date
1	Create test plan	MinhTQ	1 day	19/11/2014	19/11/2014
2	Review and complete test plan	CuongNV LamTN	1 day	21/11/2014	21/11/2014
3	Create test case	MinhTQ	1 day	24/11/2014	24/11/2014
3.1	Create test case for Android Application	MinhTQ	3 days	28/11/2014	30/11/2014
3.2	Create test case for Embedded System	MinhTQ	3 days	28/11/2014	30/11/2014
3.3	Review & Update Test case	All members	1 day	29/11/2014	29/11/2014
4	Defect log	CuongNV HoangNK	5 days	30/11/2014	4/12/2014
5	Test report	All members	1 days	6/12/2014	6/12/2014

Table 5.1: Test Schedule

5.2.3 Test Strategy

We used V model to implement testing.

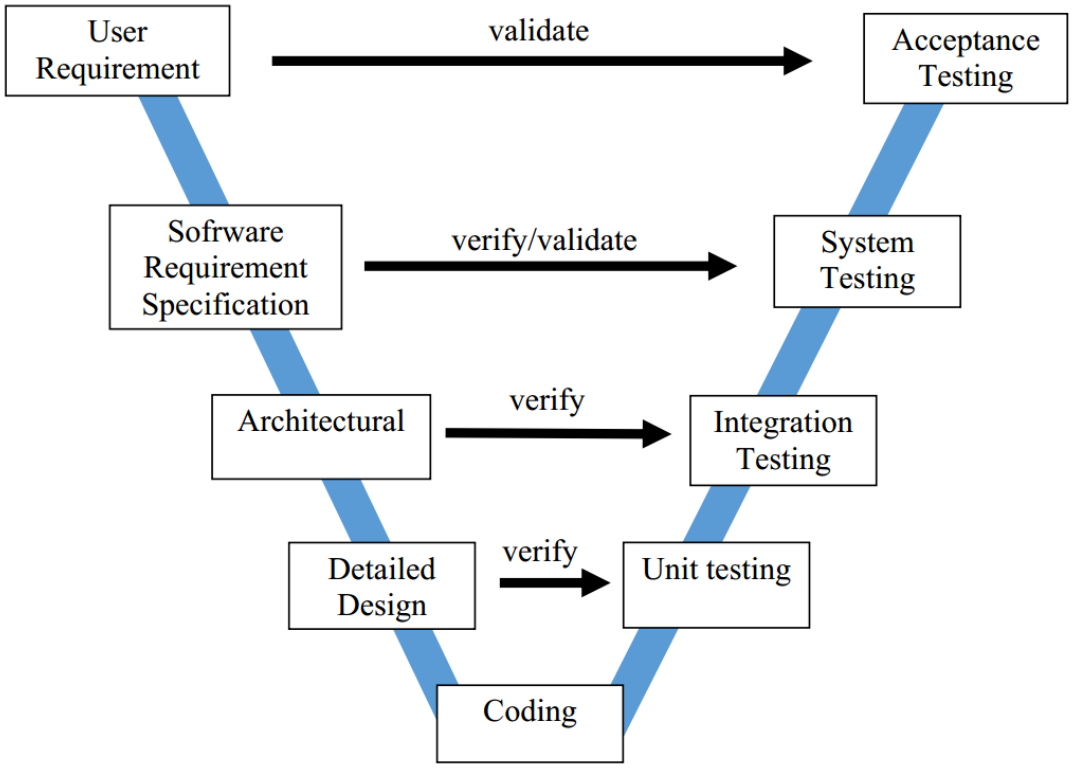


Figure 5.1: V Model

The V-model represents a software development process (also applicable to hardware development) which may be considered an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represent time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.

With V-Model, software development is separated into two appropriate phases' groups: development and testing. Testing phase is carried out simultaneously from the early of

project, the requirement specification phase. Therefore, tester could study and understand the entire requirement as well as business of project.

Moreover, there are many variations among planning, design and implementation during the whole project. To catch up with all changes and ensure logic as well as business, V-model is chose as a solution to help testers control and follow the actual workflows.

We are following phases:

Type of Tests	Stage of Test				
	Unit	System/Integration	Acceptance	Test cycle #	
				1	2
Function Test	X	X	X	X	X
User Interface test	X	X	X	X	

Table 5.2: Test Strategy

5.2.4 Testing Tools and Environment

No specific test tools are available for this project. Testers will have to setup environment and system and do actions to test.

5.3 Test Cases

5.3.1 Android Application

ID	Test Case Description	Test Case Procedure	Expected Output	Result
AA-1	In Paired Screen – To verify that displaying buttons	1. Start app 2. First screen is displayed	- Turn Off, Search and List Paired buttons are displayed.	Pass
AA-2	In Paired Screen – To verify that turning on Bluetooth on Android devices	1. Click “Search” button 2. Click “OK” button	1. Dialog box should be displayed. 2. Bluetooth is turned on.	Pass
AA-3	In Paired Screen – To verify that turning off Bluetooth on Android devices	1. Click “Turn Off” button	- On device, Bluetooth is stopped.	Fail

AA-4	In Paired Screen – To verify that displaying list of available devices that are connected with the system	1. Click “Search” button	- List of available that can be connected to the system devices is displayed.	Pass
AA-5	In Paired Screen – To verify that user enter Pin code into [PinCode] textbox.	1. Click “Search” button. 2. Choose device to connect. 3. Input Pin code into [PinCode] textbox.	1. List of devices is displayed on screen. 2. Paired Dialog will be displayed. 3. PinCode is displayed in password style in [PinCode] textbox.	Pass
AA-6	In Paired Screen – To verify that user input correct password and connect smartphone with the system	1. Click “Search” button. 2. Choose device to connect. 3. Input correct Pin Code into [PinCode] textbox. 4. Click “OK” button.	1. List of devices is displayed on screen. 2. Paired Dialog will be displayed. 3. Pin Code is display in password style in [PinCode] textbox. 4a. Smartphone is connected successfully to system. 4b. Redirect to main screen.	Fail
AA- 7	In Paired Screen – To verify that user input incorrect password and connect smartphone with the system	1. Click “Search” button. 2. Choose device to connect. 3. Input incorrect Pin Code into [PinCode] textbox. 4. Click “OK” button.	1. List of devices is displayed on screen. 2. Paired Dialog will be displayed. 3. Pin Code is display in password style in [PinCode] textbox. 4a. Error message “Pin Code is invalid” is displayed. 4b. Refresh the dialog.	Fail
AA-8	In Paired Screen – To verify that user input blank password and connect smartphone with the system	1. Click “Search” button. 2. Choose device to connect. 3. Input blank Pin Code into [PinCode] textbox. 4. Click “OK” button.	1. List of devices is displayed on screen. 2. Paired Dialog will be displayed. 3. Pin Code is display in password style in [PinCode] textbox. 4a. Error message “Pin Code is required” is displayed.	Fail

			4b. Refresh the dialog.	
AA-9	In Paired Screen – To verify that displaying list of available devices that are connected with the system	1. Click “List Paired” button	- List of devices that is connected to the system is displayed on screen.	Pass
AA-11	In Main Screen – To verify that displaying on main screen.	1. Start App 2. Pair with system successfully. 3. Main screen is display	- Main screen include: buttons, checkbox, textbox, seekbar and current Temp/Hum information	Pass
AA-12	In Main Screen – To verify that displaying current Temperature when temperature is changed on chamber	1. Information of current temperature on box is displayed.	1. Temperature is changed and displayed exactly on main screen.	Pass
AA-13	In Main Screen – To verify that displaying current Humidity when humidity is changed on box	1. Information of current humidity on box is displayed.	1. Humidity is changed and displayed exactly on main screen	Pass
AA-14	In Main Screen – To verify that displaying Temp value on Seekbar	1. Look at Temp value on Seekbar	1. Temp value must be display like last setting on SeekBar. Example: display 45ºC not 0 ºC	Pass
AA-15	In Main Screen – To verify that displaying Hum value on Seekbar	1. Look at Hum value on Seekbar	1. Hum value must be display like last setting on SeekBar. Example: display 45% not 0 ºC	Pass
AA-16	In Main Screen – To verify that setting Temp value successfully.	1. Change Temp value on Temperature Seekbar 2. Click “Set” button.	1. Temp value is changed. 2. Set and transform Temp value successfully to Arduino.	Fail
AA-17	In Main Screen – To verify that setting Hum value successfully.	1. Change Hum value on Humidity Seekbar 2. Click “Set” button.	1. Hum value is changed. 2. Set and transform Hum value successfully to Arduino..	Fail

AA-18	In Main Screen – To verify that setting Temp and Hum value successfully.	1. Change Temp value on Temperature Seekbar. 2. Change Hum value on Humidity Seekbar. 3. Click “Set” button.	1. Temp value is changed. 2. Hum value is changed. 3. Set and transform Temp and Hum value successfully to Arduino.	Pass
AA-19	In Main Screen – To verify that setting Temp value when you set Temp successfully	1. Change Temp value on Temperature Seekbar 2. Click “Set” button.	1. Temp value is changed. 2a. Set and transform Temp value successfully to Arduino. 3b. On main screen, setting Temp value is changed and displayed	Pass
AA-20	In Main Screen – To verify that setting Hum value when you set Hum successfully	1. Change Hum value on Humidity Seekbar 2. Click “Set” button.	1. Hum value is changed. 2a. Set and transform Hum value successfully to Arduino. 3b. On main screen, setting Hum value is changed and displayed	Pass
AA-21	In Main Screen – To verify that user enter command into [Command] textbox.	1. Input command into [Command] textbox.	1. Command is displayed in [Command] textbox.	Pass
AA-22	In Main Screen – To verify that clearing command in [Command] textbox.	1. Input command into [Command] textbox. 2. Click “Clear” button	1. Command is displayed in [Command] textbox. 2. Command that is entered by user is cleared in [Command] textbox.	Fail
AA-23	In Main Screen – To verify that sending command in [Command] textbox.	1. Input command into [Command] textbox. 2. Click “Send” button	1. Command is displayed in [Command] textbox. 2. Command that is entered by user is send to device setting	Pass
AA-24	In Main Screen – Check click “Disconnect” button.	1. Click “Disconnect” button.	1. Redirect to Paired Screen.	Pass

Table 5.3: Application's Test Case

5.3.2 Embedded System

ID	Test Case Description	Test Case Procedure	Expected Output	Result
ES-1	Check Bluetooth	1. Using smartphone to find and connect to Bluetooth on system	1. Smartphone can connect successfully to system	Pass
ES-2	Check RS232 communication	1. Connect the controller to PC 2. Disconnect the controller to PC	1. Data can be sent and received. 1. Data cannot be sent and received.	Pass
ES-3	Check send data from Smartphone to controller in Bluetooth	1. Connect smartphone to controller. 2. Send "aa" text from mobile to controller	1. Controller is connected to mobile via Bluetooth. 2. Using Terminal, Arduino receive "aa" text.	Fail
ES-4	Check receive data from Controller to Smartphone in Bluetooth	1. Connect smartphone to controller. 2. Using Terminal on PC, send "aa" text from Controller to Smartphone	1. Controller is connected to mobile via Bluetooth. 2. On Smartphone, Android App receives "aa" text.	Pass
ES-5	Check On/Off relay	1. When current Temp/Hum is greater than or equal to setting Temp/Hum value. 2. When current Temp/Hum is smaller than to setting Temp/Hum value.	1. Relay turn off automatic. 2. Relay turn on automatic.	Pass
ES-6	Check sensor	1. Using device that can react to temperature, humidity exactly.	1. Temperature and Humidity that are reacted by sensor should be same.	Pass

Table 5.4: Embedded System's Test Case

5.4 Test Results

First Test

Sub Module	% complete	Pass	Fail	Tester	DEV	Number of sub test cases
Android App	100%	16	8	CuongNV	LamTN	24
Embedded System	100%	5	1	CuongNV	LamTN	6
Sub total		21	9			30

Table 5.5: First Test

Final Test

Sub Module	% complete	Pass	Fail	Tester	DEV	Number of sub test cases
Android App	100%	24	0	CuongNV	LamTN	24
Embedded System	100%	6	0	CuongNV	LamTN	6
Sub total		30	0			30

Table 5.3: Final Test

CHAPTER 6: SYSTEM USER'S MANUAL

6.1. Installation Guide

6.1.1 Environment

Window 7/8/8.1

6.1.2 Installation

6.1.2.1 Install Eclipse

Go to https://dl.google.com/android/adt/adt-bundle-windows-x86_64-20140702.zip download Eclipse ADT with the Android SDK for Windows.

To set up the ADT Bundle:

1. Unpack the ZIP file (named adt-bundle-<os_platform>.zip) and save it to an appropriate location, such as a "Development" directory in your home directory.
2. Open the adt-bundle-<os_platform>/eclipse/ directory and launch **Eclipse**.

To add SDK packages:

Official adding guide: https://developer.android.com/sdk/installing/adding-packages.html

6.1.2.1 Install JDK 8

Go to http://download.oracle.com/otn-pub/java/jdk/8u25-b18/jdk-8u25-windows-x64.exe download Java SE Development Kit 8.

Updating the PATH Environment Variable

If you do not set the PATH variable, you need to specify the full path to the executable file every time you run it, such as:

```
C:\> "C:\Program Files\Java\jdk1.8.0\bin\javac" MyClass.java
```

It is useful to set the PATH variable permanently so it will persist after rebooting.

To set the PATH variable permanently, add the full path of the jdk1.8.0\bin directory to the PATH variable. Typically, this full path looks something like C:\Program Files\Java\jdk1.8.0\bin. Set the PATH variable as follows on Microsoft Windows:

1. Click **Start**, then **Control Panel**, then **System**.
2. Click **Advanced**, then **Environment Variables**.

3. Add the location of the bin folder of the JDK installation to the `PATH` variable in **System Variables**. The following is a typical value for the `PATH` variable:

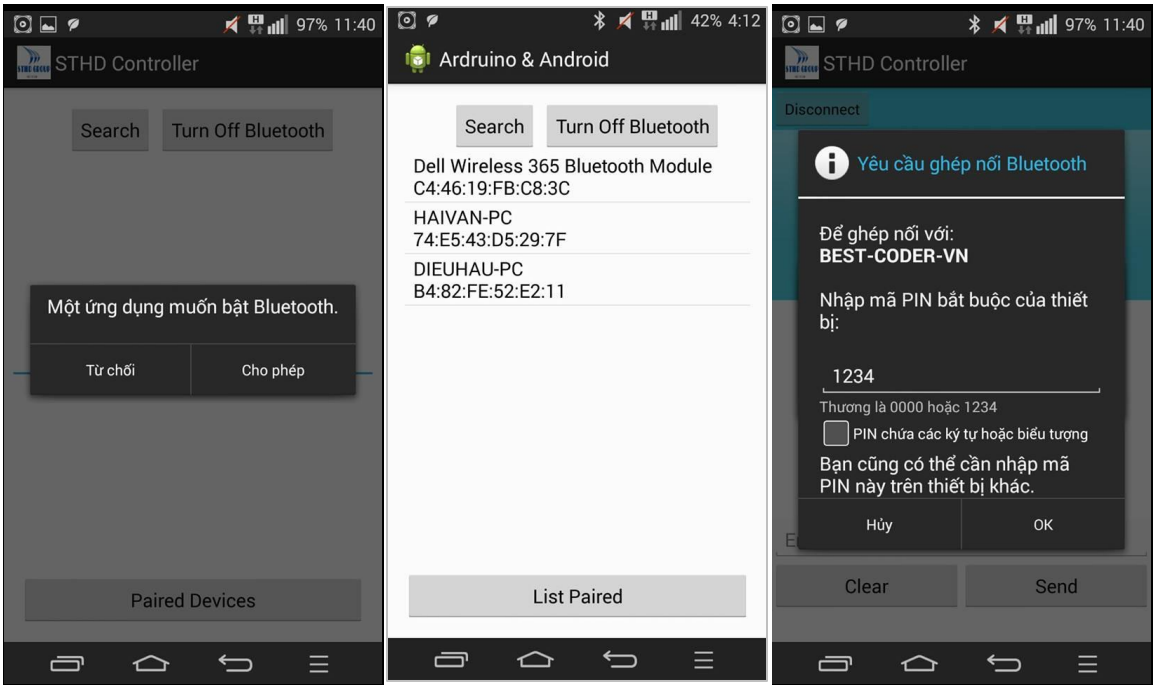
4. C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files\Java\jdk1.8.0\bin

Note:

- The `PATH` environment variable is a series of directories separated by semicolons (`;`) and is not case-sensitive. Microsoft Windows looks for programs in the `PATH` directories in order, from left to right.
- You should only have one `bin` directory for a JDK in the path at a time. Those following the first instance are ignored.
- If you are not sure where to add the JDK path, append it.
- The new path takes effect in each new command window you open after setting the `PATH` variable.

6.2 User's Guide

6.2.1 Paired Screen



Connect to the system

Step 1: Click “Search” button to list all devices that can be connected.

Step 2: Turn on the Bluetooth to connect to system.

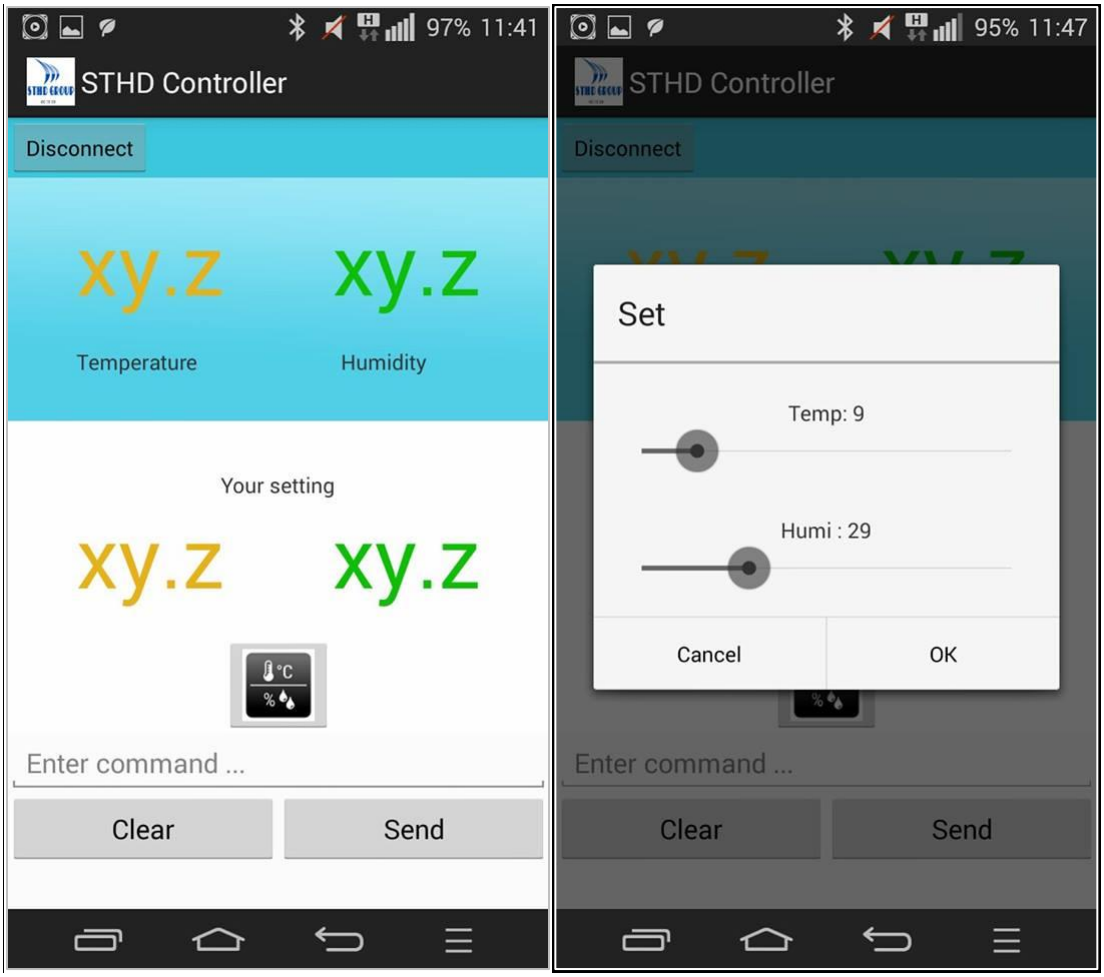
Step 3: Choose device to connect to system.

Step 4: Enter correct Pin Code and Smartphone will be connected successfully.

6.2.1 Main Screen

After that you paired successfully, Main Screen will be display on screen:

- Display current temperature and humidity on environment box.
- Display setting temperature and humidity that is set by you.
- Display warning.
- Set temperature and humidity to control Relay.
- Disconnect to system



Disconnect to system

Step 1: Click “Disconnect” button

Step 2: Redirect to paired screen and disconnect to system.

Set Temp/Hum value

Step 1: Using SeekBar to set Temp/Hum value

Step 2: Click “OK” button